

z/OS



# Network File System User's Guide



z/OS



# Network File System User's Guide

**Note**

Before using this information and the product it supports, be sure to read the general information under "Notices" on page 155.

**Third Edition, December 2001**

This edition applies to Version 1 Release 2 of z/OS™ (5694-A01) and to all subsequent releases and modifications until otherwise indicated in new editions.

Order publications through your IBM® representative or the IBM branch office serving your locality. Publications are not stocked at the address below.

IBM welcomes your comments. A form for readers' comments may be provided at the back of this publication, or you may address your comments to the following address:

International Business Machines Corporation  
RCF Processing, Department M86/050  
5600 Cottle Road  
San Jose, CA 95193-0001  
United States of America

IBMLink™ from US: STARPUBS at SJEVM5  
IBMLink from Canada: STARPUBS at TORIBM  
IBM Mail Exchange: USIB3VVD at IBMMAIL  
Internet: starpubs@us.ibm.com  
World Wide Web: <http://www.storage.ibm.com/software/sms/smshome.htm>

If you would like a reply, be sure to include your name, address, telephone number, or FAX number.

Make sure to include the following in your comment or note:

- Title and order number of this book
- Page number or topic related to your comment

When you send information to IBM, you grant IBM a nonexclusive right to use or distribute the information in any way it believes appropriate without incurring any obligation to you.

© **Copyright International Business Machines Corporation 1991, 2001. All rights reserved.**

US Government Users Restricted Rights – Use, duplication or disclosure restricted by GSA ADP Schedule Contract with IBM Corp.

# Contents

<b>Figures</b>	vii
<b>Tables</b>	ix
<b>About this book</b>	xi
Required product knowledge	xi
Where to find more information	xi
Referenced publications	xi
How to Read Syntax Diagrams	xiii
Conventions used in this book	xv
Using LookAt to look up message explanations	xvi
Accessing z/OS books on the internet	xvii
Accessing licensed books on the web	xvii
How to send your comments	xviii
Related protocol specifications	xviii
<b>Summary of changes</b>	xxi
Summary of changes for SC26-7419-02, z/OS Version 1 Release 2	xxi
New information	xxi
Changed information	xxi
Moved information	xxi
Deleted information	xxi
Summary of changes for SC26-7419-01, z/OS Version 1 Release 2	xxi
New information	xxii
Changed information	xxii
Moved information	xxii
Deleted information	xxii
Summary of changes for SC26-7419-00, Version 1 Release 1	xxii
<b>Chapter 1. Introduction</b>	1
Overview	1
Using z/OS UNIX files	2
z/OS UNIX enhancements	2
NFS protocol compliance	2
Using conventional MVS data sets	3
Mounting MVS data sets onto a client mount point	3
Crossing file systems–NFS server	3
Creating conventional MVS data sets	4
Data set serialization and sharing	4
Server control files	4
Attributes data set	4
Exports data set	5
Mount handle data set	5
Log data set	5
Checklist data set	6
Supported clients for the z/OS NFS server	6
Supported servers for the z/OS NFS client	6
WebNFS support	7
NFS version 3 and TCP protocol	7
Native ASCII support	8
<b>Chapter 2. Using conventional MVS data sets</b>	9
Special MVS considerations	9

Selecting an MVS data storage format . . . . .	9
File size determination and time stamps . . . . .	10
Ownership and permissions . . . . .	10
Statelessness . . . . .	10
Reading and writing files . . . . .	10
Case sensitivity-maplower, nomaplower . . . . .	12
Selecting text or binary processing modes-text, binary . . . . .	12
Number representation . . . . .	13
Accessing migrated files - retrieve, noretrieve; wait, nowait. . . . .	13
Accessing migrated system-managed data sets . . . . .	14
File handle refresh . . . . .	14
Mapping between the workstation and MVS file systems . . . . .	14
File name extension mapping . . . . .	15
Mounting MVS data sets onto a client mount point. . . . .	15
Using a PDS or PDSE as a directory. . . . .	16
Using multiple mount points . . . . .	17
Data set serialization and sharing . . . . .	17
NFS protocol . . . . .	17
File system size . . . . .	18
<b>Chapter 3. Creating conventional MVS data sets . . . . .</b>	<b>21</b>
Overriding data set creation attributes . . . . .	21
Preparing to create an MVS file. . . . .	21
Naming MVS files . . . . .	21
Creating physical sequential files . . . . .	22
Creating direct access files . . . . .	23
Creating PDSs and PDSEs . . . . .	23
Creating a PDS or PDSE - mkdir dsntype(pds), dsntype(library) . . . . .	24
Removing a PDS or PDSE - rm, rmdir . . . . .	24
Accessing PDS or PDSE members . . . . .	25
Updating or extending a PDS or PDSE member. . . . .	25
Timing out while writing a PDS or PDSE member . . . . .	25
Wildcard copy to a PDS or PDSE . . . . .	25
Limitations of a PDS. . . . .	26
Concurrent writes to a PDSE. . . . .	26
Creating VSAM files . . . . .	26
Exploiting SAM striped files . . . . .	27
<b>Chapter 4. Using z/OS UNIX System Services files . . . . .</b>	<b>29</b>
HFS file system . . . . .	29
Text or binary processing . . . . .	30
POSIX compatibility . . . . .	31
NFS protocol . . . . .	32
Attributes specific to z/OS UNIX System Services . . . . .	32
Synchronous write to an HFS file for NFS version 2 protocol . . . . .	32
Synchronous write to an HFS file for NFS version 3 protocol . . . . .	32
Authorization checking when writing to an HFS file. . . . .	33
HFS site attribute . . . . .	33
Protecting your z/OS UNIX System Services files . . . . .	33
Accessing z/OS UNIX System Services files from a client . . . . .	33
Mount examples . . . . .	34
Linking an MVS data set to a hierarchical file system . . . . .	34
Creating an external link . . . . .	34
Displaying the contents on an external link. . . . .	35
Deleting an external link . . . . .	35
UNIX look and feel . . . . .	35

File system size . . . . .	36
<b>Chapter 5. Network Lock Manager (NLM) and Network Status Monitor (NSM)</b> . . . . .	37
Using Network Lock Manager (NLM) . . . . .	37
Monitored locks. . . . .	37
Non-monitored locks . . . . .	37
Locking records . . . . .	37
Locking a file . . . . .	38
Using Network Status Monitor (NSM). . . . .	38
<b>Chapter 6. Commands and examples for AIX and UNIX clients</b> . . . . .	39
Using commands on AIX . . . . .	39
Quick reference of AIX and UNIX commands . . . . .	40
Accessing z/OS UNIX System Services and conventional MVS . . . . .	41
Mvslogin command examples . . . . .	41
Mount command examples . . . . .	43
Using mount to override server default attributes . . . . .	43
Displaying default and mount point attributes - showattr . . . . .	44
Unmounting and logging out of MVS . . . . .	47
Disconnecting your mount point - umount . . . . .	47
Ending your MVS session - mvslogout, mvslogut . . . . .	48
<b>Chapter 7. Commands and examples for Sun PC-NFS clients</b> . . . . .	49
Command summary for Sun PC-NFS users . . . . .	49
Mvslogin example . . . . .	50
Net use example . . . . .	50
Showattr example . . . . .	51
Net use /d example . . . . .	51
Mvslogut example . . . . .	51
<b>Chapter 8. Commands and examples for z/OS NFS clients</b> . . . . .	53
Using commands on DFSMS/MVS . . . . .	53
Accessing MVS. . . . .	55
Mvslogin command examples . . . . .	56
Mount command syntax and examples . . . . .	57
Displaying client statistical information-nfsstat . . . . .	62
Displaying server mount information-showmount . . . . .	65
Displaying default and mount point attributes-showattr . . . . .	67
Unmounting and logging out of MVS . . . . .	68
<b>Chapter 9. Attributes for the z/OS NFS client</b> . . . . .	71
Client attributes. . . . .	71
DataCaching attribute . . . . .	73
Mount processing parameters and installation parameters . . . . .	74
<b>Chapter 10. Attributes for the z/OS NFS server</b> . . . . .	75
Attributes used for z/OS UNIX System Services file access . . . . .	75
Using multipliers . . . . .	75
Specifying attributes multiple times . . . . .	75
Data set creation attributes . . . . .	76
Processing attributes. . . . .	78
Timeout attributes . . . . .	82
Retrieve attributes. . . . .	83
Mapped keyword processing attribute . . . . .	83
Native ASCII processing attributes. . . . .	84

Site attributes . . . . .	86
<b>Appendix A. Network File System messages . . . . .</b>	<b>93</b>
Server messages . . . . .	93
Client messages . . . . .	118
Messages from the client platform (AIX) . . . . .	135
<b>Appendix B. Return codes . . . . .</b>	<b>139</b>
<b>Appendix C. Handling of the file size value . . . . .</b>	<b>143</b>
Storage of the file size value . . . . .	143
System-managed PS, VSAM, and PDSE data sets . . . . .	143
Migrated system-managed data sets . . . . .	143
Non-system-managed, PDS, and direct access data sets . . . . .	144
How the file size value is generated. . . . .	144
Using fastfilesize to avoid read-for-xize . . . . .	144
Nofastfilesize . . . . .	145
<b>Appendix D. Handling of the time stamps . . . . .</b>	<b>147</b>
Time stamps for system-managed VSAM and PS data sets . . . . .	147
Time stamps for non-system-managed PS and DA data sets . . . . .	147
Storing time stamps . . . . .	147
Client program requirements . . . . .	147
Generating time stamps . . . . .	147
Time stamps for non-system-managed VSAM data sets . . . . .	148
Time stamps for PDSs and PDSEs . . . . .	148
Setting time stamps. . . . .	149
<b>Appendix E. Sample program for lockd and statd . . . . .</b>	<b>151</b>
<b>Notices . . . . .</b>	<b>155</b>
Programming interface information . . . . .	156
Trademarks. . . . .	156
<b>Glossary . . . . .</b>	<b>157</b>
<b>Index . . . . .</b>	<b>165</b>



---

## Figures

1. NFS client-server relationship . . . . .	1
2. Examples of mounting MVS data sets on OS/2, DOS, and UNIX clients . . . . .	16
3. Example of mounting an HFS file from a UNIX client . . . . .	30
4. Syntax for commands . . . . .	39
5. Displaying default attributes . . . . .	45
6. Displaying default and mount point attributes . . . . .	46
7. TSO mount command syntax operands . . . . .	57
8. Displaying NFS client rpc and NFS statistical information . . . . .	63
9. Displaying NFS client rpc statistical information. . . . .	63
10. Displaying NFS client NFS statistical information . . . . .	64
11. Displaying NFS mounted file system information . . . . .	64
12. Displaying V.2 protocol NFS mounted file system information . . . . .	65
13. Displaying version 3 protocol NFS mounted file system information . . . . .	65
14. Displaying server mount hostname and directory information. . . . .	65
15. Displaying server mount directory information . . . . .	66
16. Displaying server mount hostname information . . . . .	66
17. Displaying server export information. . . . .	66
18. Displaying server export information, security(none) . . . . .	66
19. Displaying server export information, security(safexp) . . . . .	67
20. Displaying default attributes . . . . .	68
21. TSO unmount command syntax operands . . . . .	69
22. Sample program for record locking and file locking . . . . .	151



# Tables

1. Additional information . . . . .	xi
2. Conventions used in this book . . . . .	xv
3. View of NFS server capability . . . . .	7
4. Breakdown of text and binary writes . . . . .	11
5. NFS procedures . . . . .	17
6. File system values to get dynamic file system information (fsstat) . . . . .	18
7. File system values to get static file system information (fsinfo) . . . . .	18
8. File system values to retrieve POSIX information (pathconf) . . . . .	18
9. Examples of the mount command for clients . . . . .	34
10. File system values to get static file system information (fsinfo) . . . . .	36
11. File system values to retrieve POSIX information (pathconf) . . . . .	36
12. Examples of the mvslogin command for clients . . . . .	41
13. Examples of the showattr command for clients . . . . .	45
14. Examples of the umount command for clients . . . . .	47
15. Example of the mvslogout command for clients . . . . .	48
16. Examples of the mvslogin command for clients . . . . .	56
17. Mount operand . . . . .	57
18. Attributes - z/OS NFS client . . . . .	71
19. Client attributes . . . . .	71
20. Mount processing parameters . . . . .	74
21. Installation parameters . . . . .	74
22. Attributes - z/OS NFS server . . . . .	75
23. Data set creation attributes . . . . .	76
24. Processing attributes . . . . .	78
25. The mapped keyword and existing keywords . . . . .	83
26. z/OS NFS clients with non-390 based NFS servers . . . . .	85
27. NFS servers with non-390 based NFS clients . . . . .	85
28. Site attributes . . . . .	87
29. Messages - client operating system, NFS server, and NFS client . . . . .	93
30. NFS server log data set message format . . . . .	93
31. NFS client MVS operators console message format . . . . .	118
32. NFS client log data set message format . . . . .	119
33. Common variables . . . . .	120
34. Parsing error (when reason code is 6E01xxxx) . . . . .	120
35. Parsing error (when reason code is from 6E0111yy to 6E0129yy) . . . . .	121
36. Parsing error (when reason code is 6Exxyyyy') . . . . .	121
37. GFSC845I valid parameters . . . . .	134
38. Externalized return codes defined by the NFS version 2 protocol . . . . .	139
39. Externalized return codes defined by the NFS version 3 protocol . . . . .	139
40. z/OS NFS version 1 protocol externalized return code mapping to z/OS UNIX . . . . .	141
41. z/OS NFS return codes between NFS version 3 protocol and z/OS UNIX . . . . .	141



---

## About this book

This book provides the z/OS Network File System (NFS) client user and the administrator with information about using the z/OS NFS, and includes the following components:

- An introduction to the z/OS NFS
- A description and examples of the commands that are used to access data sets on the MVS host processor, and a description and examples of the commands that are used to display and override the data set (file) attributes
- A description of how to create your own data sets on the host processor, with descriptions (and examples) of the required attributes
- A description and examples of the z/OS NFS client commands that are used to access data on systems which support the NFS protocols. The remote NFS server can be an MVS, UNIX®, AIX® or other system
- A description of how the z/OS NFS Network Lock Manager (NLM) and the z/OS NFS Network Status Monitor (NSM) run with z/OS NFS
- Explanations of the messages that are sent to the z/OS NFS client
- A list of related protocol specifications
- Information on handling the file size value
- Information on handling the time stamps
- A list of acronyms and definitions of terms that are used in this book

In this book, the z/OS NFS server is also referred to as the server.

---

## Required product knowledge

If you are creating or accessing conventional MVS data sets, you should be familiar with the MVS file system. You should also be familiar with the operating environment of the system that you are using to run the Network File System (NFS) client software.

---

## Where to find more information

Where necessary, this book references information in other books, using the shortened version of the book title. For complete titles and order numbers of the books for all products that are part of z/OS, see *z/OS Information Roadmap*.

---

## Referenced publications

The following publications have additional information:

*Table 1. Additional information*

Publication Title	Order Number
<i>AIX for RISC System/6000 Performance Monitoring and Tuning Guide</i>	SC23-2365
<i>AIX Operating System TCP/IP User's Guide</i>	SC23-2309
<i>AIX Commands Reference for RISC System/6000, Volume 1</i>	GC23-2376
<i>AIX Commands Reference for RISC System/6000, Volume 2</i>	GC23-2366
<i>AIX Commands Reference for RISC System/6000, Volume 3</i>	GC23-2367
<i>AIX Commands Reference for RISC System/6000, Volume 4</i>	GC23-2393

Table 1. Additional information (continued)

<b>Publication Title</b>	<b>Order Number</b>
<i>AIX Communications Concepts and Procedures for IBM RISC System/6000</i>	GC23-2203
<i>Character Data Representation Architecture Reference and Registry</i>	SC09-2190
<i>Character Data Representation Architecture Overview</i>	GC09-2207
<i>DFSMS/MVS Online Product Library</i>	LK2T-8731
<i>IBM Online Library: AIX Base Collection Kit</i>	SK2T-2166
<i>IBM Online Library Omnibus Edition: MVS Collection</i>	SK27-0710
<i>IBM Networking Softcopy Collection Kit</i>	SK2T-6012
<i>Introducing IBM's Transmission Control Protocol/Internet Protocol Products for OS/2®, MVS, VM</i>	GC31-6080
<i>OS/390 Support for Unicode™: Using Conversion Services</i>	SC33-7050
<i>SMP/E R8.1 Reference</i>	SC28-1107
<i>TCP/IP: Performance Tuning Guide</i>	SC31-7188
<i>TCP/IP for MVS: Programmer's Reference</i>	SC31-7135
<i>TCP/IP for MVS: Customization and Administration Guide</i>	SC31-7134
<i>TCP/IP for MVS: User's Guide</i>	SC31-7136
<i>z/OS DFSMSdfp Advanced Services</i>	SC26-7400
<i>z/OS DFSMSdfp Diagnosis Reference</i>	GY27-7618
<i>z/OS DFSMSdfp Storage Administration Reference</i>	SC26-7402
<i>z/OS DFSMSshm Diagnosis Guide</i>	LY35-0114
<i>z/OS MVS Programming: Authorized Assembler Services Guide</i>	SA22-7608
<i>z/OS MVS Programming: Authorized Assembler Services Reference ALE-DYN ALE-DYN</i>	SA22-7609
<i>z/OS MVS Programming: Authorized Assembler Services Reference ENF-IXG ENF-IXG</i>	SA22-7610
<i>z/OS MVS Programming: Authorized Assembler Services Reference LLA-SDU LLA-SDU</i>	SA22-7611
<i>z/OS MVS Programming: Authorized Assembler Services Reference SET-WTO SET-WTO</i>	SA22-7612
<i>z/OS DFSMS Introduction</i>	SC26-7397
<i>z/OS DFSMS Macro Instructions for Data Sets</i>	SC26-7408
<i>z/OS DFSMS Migration</i>	GC26-7398
<i>z/OS DFSMS: Using Data Sets</i>	SC26-7410
<i>z/OS MVS System Management Facilities (SMF)</i>	SA22-7630
<i>z/OS MVS Installation Exits</i>	SA22-7593
<i>z/OS MVS JCL Reference</i>	SA22-7597
<i>z/OS MVS System Messages, Vol 1 (ABA-ASA)</i>	SA22-7631
<i>z/OS MVS System Messages, Vol 2 (ASB-EWX)</i>	SA22-7632
<i>z/OS MVS System Messages, Vol 3 (GDE-IEB)</i>	SA22-7633
<i>z/OS MVS System Messages, Vol 4 (IEC-IFD)</i>	SA22-7634
<i>z/OS MVS System Messages, Vol 5 (IGD-IZP)</i>	SA22-7635
<i>z/OS MVS Programming: Authorized Assembler Services Guide</i>	SA22-7608

Table 1. Additional information (continued)

Publication Title	Order Number
<i>z/OS Communications Server IP Configuration Reference</i>	SC31-8776
<i>z/OS Network File System Customization and Operation</i>	SC26-7417
<i>z/OS Network File System Performance Tuning Guide</i>	SC26-7418
<i>z/OS TSO/E User's Guide</i>	SA22-7794
<i>z/OS UNIX System Services File System Interface Reference</i>	SA22-7808
<i>z/OS Information Roadmap</i>	SA22-7500
<i>z/OS UNIX System Services Messages and Codes</i>	SA22-7807
<i>z/OS SecureWay Security Server RACF Security Administrator's Guide</i>	SA22-7683
<i>z/OS SecureWay Security Server RACF System Programmer's Guide</i>	SA22-7681
<i>z/OS SMP/E User's Guide</i>	SA22-7773
<i>z/OS UNIX System Services Programming: Assembler Callable Services Reference</i>	SA22-7803
<i>z/OS UNIX System Services Programming Tools</i>	SA22-7805
<i>z/OS UNIX System Services Command Reference</i>	SA22-7802
<i>z/OS UNIX System Services User's Guide</i>	SA22-7801
<i>z/OS UNIX System Services Planning</i>	GA22-7800

## How to Read Syntax Diagrams

Throughout this library, diagrams are used to illustrate the programming syntax. Keyword parameters are parameters that follow the positional parameters. Unless otherwise stated, keyword parameters can be coded in any order. The following list tells you how to interpret the syntax diagrams:

- Read the diagrams from left-to-right, top-to-bottom, following the main path line. Each diagram begins on the left with double arrowheads and ends on the right with two arrowheads facing each other.

►► | Syntax Diagram | ◄◄

- If a diagram is longer than one line, each line to be continued ends with a single arrowhead and the next line begins with a single arrowhead.

►► | First Line | ◄

►► | Second Line | ◄

►► | Last Line | ◄◄

- Required keywords and values appear on the main path line. You must code required keywords and values.

►►—REQUIRED\_KEYWORD—◄◄

If several mutually exclusive required keywords or values exist, they are stacked vertically in alphanumeric order.

►►—  
 └─REQUIRED\_KEYWORD\_OR\_VALUE\_1─  
 └─REQUIRED\_KEYWORD\_OR\_VALUE\_2─◄◄

- Optional keywords and values appear below the main path line. You can choose not to code optional keywords and values.

►►—  
 └─KEYWORD─◄◄

If several mutually exclusive optional keywords or values exist, they are stacked vertically in alphanumeric order below the main path line.

►►—  
 └─KEYWORD\_OR\_VALUE\_1─  
 └─KEYWORD\_OR\_VALUE\_2─◄◄

- An arrow returning to the left above a keyword or value on the main path line means that the keyword or value can be repeated. The comma means that each keyword or value must be separated from the next by a comma.

►►—  
 └─  
 └─,─  
 └─REPEATABLE\_KEYWORD─◄◄

- An arrow returning to the left above a group of keywords or values means more than one can be selected, or a single one can be repeated.

►►—  
 └─  
 └─,─  
 └─  
 └─REPEATABLE\_KEYWORD\_OR\_VALUE\_1─  
 └─REPEATABLE\_KEYWORD\_OR\_VALUE\_2─◄◄

- A word in all uppercase is a keyword or value you must spell exactly as shown. In this example, you must code **KEYWORD**.

►►—KEYWORD—◄◄

If a keyword or value can be abbreviated, the abbreviation is discussed in the text associated with the syntax diagram.

- If a diagram shows a character that is not alphanumeric (such as parentheses, periods, commas, and equal signs), you must code the character as part of the syntax. In this example, you must code **KEYWORD=(001,0.001)**.



►►—KEYWORD=(001,0.001)—————◄◄

- If a diagram shows a blank space, you must code the blank space as part of the syntax. In this example, you must code **KEYWORD=(001 FIXED)**.

►►—KEYWORD=(001 FIXED)—————◄◄

- Default keywords and values appear above the main path line. If you omit the keyword or value entirely, the default is used.

►►—DEFAULT  
KEYWORD—————◄◄

- A word in all lowercase italics is a *variable*. Where you see a variable in the syntax, you must replace it with one of its allowable names or values, as defined in the text.

►►—*variable*—————◄◄

- References to syntax notes appear as numbers enclosed in parentheses above the line. Do not code the parentheses or the number.

(1)  
►►—KEYWORD—————◄◄

### Notes:

1 An example of a syntax note.

- Some diagrams contain *syntax fragments*, which serve to break up diagrams that are too long, too complex, or too repetitious. Syntax fragment names are in mixed case and are shown in the diagram and in the heading of the fragment. The fragment is placed below the main diagram.

►►—| Reference to Syntax Fragment |—————◄◄

### Syntax Fragment:

|—1ST\_KEYWORD,2ND\_KEYWORD,3RD\_KEYWORD—————|

---

## Conventions used in this book

Table 2 shows conventions used in this book:

Table 2. Conventions used in this book

, Commas are used as separators when specifying data set attributes.

Table 2. Conventions used in this book (continued)

( )	Parantheses in data set attributes must be passed to MVS, but the AIX or UNIX client systems' shells recognize parantheses and strip them off. To prevent this, use one of these methods: <ul style="list-style-type: none"><li>• Include the attributes in double (") or single (') quotation marks</li><li>• Include each parenthetical expression in quotation marks</li><li>• Precede each parenthesis with a back slash ( \ )</li></ul>
\$	The dollar sign can be used in the file name of an MVS mounted data set on UNIX. To prevent UNIX from trying to interpret \$ as part of a variable, use one of these methods: <ul style="list-style-type: none"><li>• Include the entire file name in single (') quotation marks (for example, vi '\$file2')</li><li>• Precede each file name with a back slash ( \ ) (for example, vi \ \$file2).</li></ul> <p>The dollar sign is also used as a prompt in the screen examples indicating that any user can enter the command.</p>

Prompts for users and superusers vary depending on the original AIX or UNIX implementation, or on site defaults. This book follows the \$ and # conventions outlined above for illustration.

This book uses a "railroad track" syntax convention to show how to enter commands. To clarify the explanations, both railroad track diagrams and screen samples are used when illustrating commands.

Commands for AIX and UNIX implementations are case sensitive. NFS commands issued at the client are shown in lowercase letters.

---

## Using LookAt to look up message explanations

LookAt is an online facility that allows you to look up explanations for z/OS messages, system abends, and some codes. Using LookAt to find information is faster than a conventional search because in most cases LookAt goes directly to the message explanation.

You can access LookAt from the Internet at:

<http://www.ibm.com/servers/eserver/zseries/zos/bkserv/lookat/lookat.html>

or from anywhere in z/OS where you can access a TSO command line (for example, TSO prompt, ISPF, z/OS UNIX System Services running OMVS).

To find a message explanation on the Internet, go to the LookAt Web site and simply enter the message identifier (for example, IAT1836 or IAT\*). You can select a specific release to narrow your search. You can also download code from the *IBM Online Library Omnibus Edition z/OS Collection*, SK2T-6700 and the LookAt Web site so you can access LookAt from a PalmPilot (Palm VIIx suggested).

To use LookAt as a TSO command, you must have LookAt installed on your host system. You can obtain the LookAt code for TSO from a disk on your *IBM Online Library Omnibus Edition z/OS Collection*, SK2T-6700 or from the LookAt Web site. To obtain the code from the LookAt Web site, do the following:

1. Go to <http://www.ibm.com/servers/eserver/zseries/zos/bkserv/lookat/lookat.html>.
2. Click the **News** button.
3. Scroll to **Download LookAt Code for TSO and VM**.

4. Click the ftp link, which will take you to a list of operating systems. Select the appropriate operating system. Then select the appropriate release.
5. Find the **lookat.me** file and follow its detailed instructions.

To find a message explanation from a TSO command line, simply enter: **lookat message-id**. LookAt will display the message explanation for the message requested.

**Note:** Some messages have information in more than one book. For example, IEC192I has routing and descriptor codes listed in *z/OS MVS Routing and Descriptor Codes*. For such messages, LookAt prompts you to choose which book to open.

---

## Accessing z/OS books on the internet

In addition to making softcopy books available on CD-ROM, IBM provides access to unlicensed z/OS softcopy books on the Internet. To find z/OS books on the Internet, first go to the z/OS home page:

<http://www.ibm.com/servers/eserver/zseries/zos/>

From this Web site, you can link directly to the z/OS softcopy books by selecting the **Library** icon. You can also link to IBM Direct to order hardcopy books.

---

## Accessing licensed books on the web

z/OS licensed documentation in PDF format is available on the Internet at the IBM Resource Link Web site at:

<http://www.ibm.com/servers/resourceLink>

Licensed books are available only to customers with a z/OS license. Access to these books requires an IBM Resource Link Web userid and password, and a key code. With your z/OS order you received a memo that includes this key code.

To obtain your IBM Resource Link Web userid and password, log on to:

<http://www.ibm.com/servers/resourceLink>

To register for access to the z/OS licensed books, follow these steps:

1. Log on to Resource Link using your Resource Link userid and password.
2. Click on **User Profiles** located on the left-hand navigation bar.
3. Click on **Access Profile**.
4. Click on **Request Access to Licensed books**.
5. Supply your key code where requested and click on the **Submit** button.

If you supplied the correct key code you will receive confirmation that your request is being processed. After your request is processed you will receive an e-mail confirmation.

**Note:** You cannot access the z/OS licensed books unless you have registered for access to them and received an e-mail confirmation informing you that your request has been processed.

To access the licensed books, follow these steps:

1. Log on to Resource Link using your Resource Link userid and password.

2. Click on **Library**.
3. Click on **zSeries**.
4. Click on **Software**.
5. Click on **z/OS**.
6. Access the licensed book by selecting the appropriate element.

---

## How to send your comments

Your feedback is important in helping to provide the most accurate and high-quality information. If you have any comments about this book or other NFS documentation, you can select one of the following options:

- Visit our home page at:  
<http://www.ibm.com/servers/eserver/zseries/zos/nfs/>

From this Web site you can click on the **Contact Us** icon to enter and submit your comments, or click on the **Library** icon to access NFS books.

- Send your comments about this book, other NFS books, or any DFSMS documentation by e-mail to:
  - IBMLink from US: [starpubs@us.ibm.com](mailto:starpubs@us.ibm.com)
  - IBMLink from Canada: STARPUBS at TORIBM
  - IBM Mail Exchange: USIB3VVD at IBMMAIL
  - Internet: [starpubs@us.ibm.com](mailto:starpubs@us.ibm.com)

Be sure to include the name of the book, the part number of the book, version and product name, and if applicable, the specific location of the text you are commenting on (for example, a page number or a table number).

- Fill out one of the forms at the back of this book and return it by mail or by giving it to an IBM representative. If the form has been removed, address your comments to IBM Corporation, RCF Processing Department M86/050, 5600 Cottle Road, San Jose, California 95193-0001, U.S.A.

---

## Related protocol specifications

IBM is committed to industry standards. The internet protocol suite is still evolving through Requests for Comments (RFC). New protocols are being designed and implemented by researchers, and are brought to the attention of the internet community in the form of RFCs. Some of these are so useful that they become a recommended protocol. That is, all future implementations for TCP/IP are recommended to implement this particular function or protocol. These become the *de facto* standards on which the TCP/IP protocol suite is built.

The NFS is implemented as a set of RPC procedures that use External Data Representation (XDR) encoding to pass arguments between clients and servers. The NFS is based on the following RFCs:

Internet Protocol	RFC 791, J.B. Postel
NFS: Network File System Version 2 Protocol Specification	RFC 1094, Sun Microsystems, Incorporated
NFS: Network File System Version 3 Protocol Specification	RFC 1813, Sun Microsystems, Incorporated
Open Group Technical Standard Protocols for Interworking: XNFS, Version 3W	Document Number: C702
RPC: Remote Procedure Call Protocol Specification Version 2	RFC 1057, Sun Microsystems Incorporated

RPC: Remote Procedure Call Protocol Specification Version 2	RFC 1831, R. Srinivasan
User Datagram Protocol	RFC 768, J.B. Postel
WebNFS Client Specification	RFC 2054, B. Callaghan
WebNFS Server Specification	RFC 2055, B. Callaghan
XDR: External Data Representation Standard	RFC 1014, Sun Microsystems, Incorporated
XDR: External Data Representation Standard	RFC 1832, R. Srinivasan

For more information about Request for Comments (RFC), see the Internet Engineering Task Force (IETF) home page:

<http://www.ietf.org/>

RFC documentation is available at the Internet Engineering Task Force (IETF) Web site:

<http://www.ietf.org/rfc.html>



---

## Summary of changes

The summary of changes informs you of changes to this book. Revision bars (|) in the left margin of the book indicate changes from the previous edition.

---

### Summary of changes for SC26-7419-02, z/OS Version 1 Release 2

This book contains information that was previously presented in *z/OS Network File System User's Guide*, SC26-7419-01. The following sections summarize the changes to that information.

#### New information

This edition includes new information:

- New message information; see “Server messages” on page 93.
- New site attribute **hfsfbtimeout**; see “Site attributes” on page 86.
- Additional reference information; see Table 1 on page xi.

#### Changed information

This edition includes changed information:

- (<PROCNAME>) in messages; see “Server messages” on page 93.
- Locating message information; see “Using LookAt to look up message explanations” on page xvi.
- Site attribute **bufhigh** *n* integer value; see “Site attributes” on page 86.

#### Moved information

This edition does not include moved information:

#### Deleted information

This edition deleted the server message (GFSA905I) information from “Server messages” on page 93.

This book contains terminology, maintenance, and editorial changes. Technical changes or additions to the text and illustrations are indicated by a vertical line to the left of the change.

Starting with z/OS V1R2, you may notice changes in the style and structure of some content in this book. For example, headings that use uppercase for the first letter of initial words only, and text that has a different look and format. The changes are ongoing improvements to the consistency and retrievability of information in our books.

---

### Summary of changes for SC26-7419-01, z/OS Version 1 Release 2

**Attention:** This was a major revision of the Network File System (NFS) features in OS/390® (5647-A01 (V2R6, V2R7, V2R8, V2R9, V2R10)) and z/OS (5694-A01 (V1R1, V1R2)); it provides information for NFS features that are applicable to these releases.

This book contains information that was previously presented in *z/OS Network File System User's Guide*, SC26-7419-00. The following sections summarize the changes to that information.

## New information

This edition includes the following new information:

- Network File System (NFS) features; see “Chapter 1. Introduction” on page 1.
- Mixing file systems; see “Crossing file systems–NFS server” on page 3.
- Reason codes; see Table 36 on page 121.
- NFS client attributes; see “Chapter 9. Attributes for the z/OS NFS client” on page 71.
- The attribute **TAG(TEXT,CCSID)**; see Table 17 on page 57.
- TCP transport including a new parameter **proto(tcp/udp)**. The proto parameter lets the NFS client select the transport for communication with another NFS server. See “Client attributes” on page 71.
- Processing attributes **mapped**, **rdrverf**, **nordrverf**, **cln\_ccsid(n)**, and **srv\_ccsid(n)**; see “Processing attributes” on page 78, and “Mapped keyword processing attribute” on page 83.
- The site attribute **readdirtimeout**; see “Site attributes” on page 86.
- Included NFS server messages GFSA320 through GFSA950; see “Server messages” on page 93.
- Included NFS client messages GFSC100 through GFSC840; see “Client messages” on page 118.
- Included new glossary terms; see “Glossary” on page 157.

## Changed information

This edition includes the following changed information:

- Reading and writing data; see “Data conversion” on page 58.
- As part of the name change of OpenEdition® to z/OS UNIX System Services, occurrences of z/OS OpenEdition have been changed to z/OS UNIX System Services or its abbreviated name, z/OS UNIX. OpenEdition may continue to appear in messages, panel text, and other code with z/OS UNIX System Services.

## Moved information

This edition includes the following moved information:

- Client attributes; see “Chapter 9. Attributes for the z/OS NFS client” on page 71.
- Comment requests; see “Related protocol specifications” on page xviii.

## Deleted information

This edition deletes the following information:

- OS/2 references are removed; however, some OS/2 references may continue to display in example text.
- IBM TCP/IP for DOS references are removed; however, some DOS references may continue to display in example text.
- *Commands and Examples for OS/2 Clients* information was removed.

This book contains terminology, maintenance, and editorial changes, including changes to improve consistency and retrievability.

---

## Summary of changes for SC26-7419-00, Version 1 Release 1

This book applies to Version 1 Release 1 of z/OS and contains information that was previously presented in *OS/390 Network File System User's Guide*, SC26-7254.



# Chapter 1. Introduction

This chapter explains the NFS client-server relationship and introduces the IBM Network File System (z/OS NFS). When used to access z/OS UNIX System Services (z/OS UNIX) data, which conforms to portable operating system interface (POSIX) standards, it is similar to other UNIX/AIX NFS systems.

## Overview

A client is a computer or process that requests services on the network. A server is a computer or process that responds to a request for service from a client. A user accesses a service, which allows the use of data or other resources.

Figure 1 illustrates the client-server relationship. The upper right portion of the figure shows the z/OS NFS server. The lower right portion of the figure shows the z/OS NFS client. The left portion of the figure shows various NFS clients and servers which can interact with the z/OS NFS server and client. The center of the figure shows the Transmission Control Protocol/Internet Protocol (TCP/IP) network used to communicate between the clients and servers.

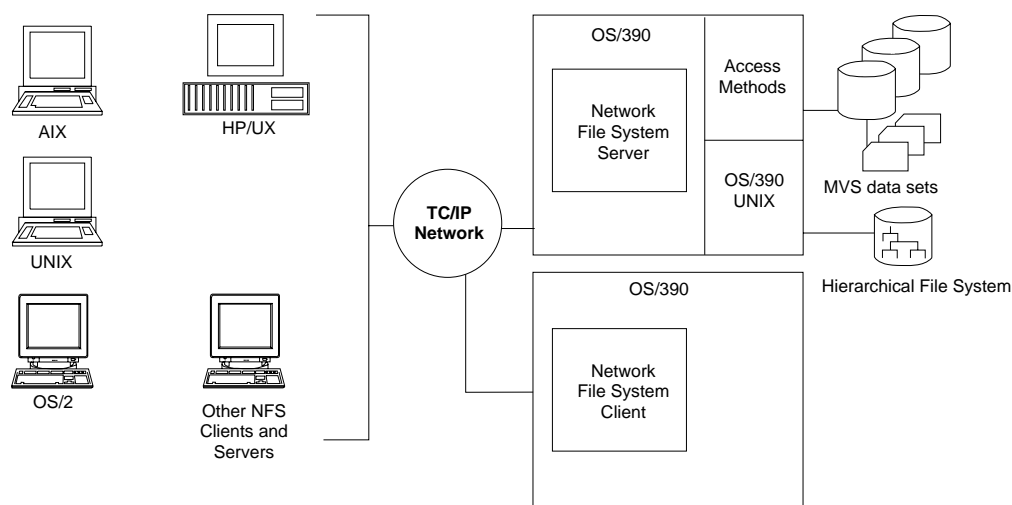


Figure 1. NFS client-server relationship

With the z/OS NFS server, you can remotely access MVS/ESA™ conventional data sets or z/OS UNIX files from workstations, personal computers, and other systems that run client software for the Sun NFS version 2 protocols, the Sun NFS version 3 protocols, and the WebNFS protocols over TCP/IP network.

The z/OS NFS server acts as an intermediary to read, write, create or delete z/OS UNIX files and MVS data sets that are maintained on an MVS host system. The remote MVS data sets or z/OS UNIX files are mounted from the host processor to appear as local directories and files on the client system. This server makes the strengths of an MVS host processor (storage management, high-performance disk storage, security, and centralized data) available to the client platforms.

With the z/OS NFS client you can allow basic sequential access method (BSAM), queued sequential access method (QSAM), virtual storage access method (VSAM), and z/OS UNIX users and applications transparent access to data on systems

which support the Sun NFS version 2 protocols and the Sun NFS version 3 protocols. The remote NFS Server can be an MVS, UNIX, AIX, or other system. The z/OS NFS client is implemented on z/OS UNIX and implements the client portion of the Sun NFS version 2 protocols and the Sun NFS version 3 Protocols.

The NFS uses the communication services provided by TCP/IP, a suite of protocols that includes the remote procedure call (RPC) and External Data Representation (XDR) protocols. RPC allows a program on one machine to start a procedure on another machine, as if the procedure is local. XDR resolves the differences in data representation of different machines.

The NFS, then, can be used for file sharing between platforms and file serving (as a data repository).

If you are using NFS as a file server, the hierarchical file system (HFS) might be a better choice than using conventional MVS data sets, because of its UNIX-like features.

---

## Using z/OS UNIX files

The NFS server enables the client user remote access to z/OS UNIX files from a client workstation.

z/OS UNIX provides a hierarchical file system (HFS) for MVS. The HFS file system is similar to a UNIX file system. All z/OS UNIX files reside in a directory, which in turn is a file in a higher level directory. The highest level directory is called the root directory.

When client users mount files from your server system, you use a common HFS prefix to distinguish z/OS UNIX files from conventional MVS data sets. You see z/OS UNIX files in a standard UNIX format on your workstation, but the files are stored on an MVS host system.

Using the NFS, the client can mount all or part of the z/OS UNIX file system and make it appear as part of your local file system. From there the client user can create, delete, read, write, and treat the host-located files as part of the workstation's own file system. For more information about z/OS UNIX see *z/OS UNIX System Services User's Guide*.

## z/OS UNIX enhancements

z/OS UNIX file system support provides these enhancements over conventional MVS data sets:

- Support for hierarchical directories
- File names up to 255 characters in length
- Path names up to 1023 characters in length
- Mixed case names and special characters, except nulls and slash characters, in file and path names
- UNIX-style access permission support
- Group and user ID support at a file level
- Ability to link conventional MVS data sets to a POSIX path name

## NFS protocol compliance

The NFS provides full NFS protocol compliance for accessing the HFS file system.

---

## Using conventional MVS data sets

Using the NFS, you can access conventional MVS data sets from a client workstation, personal computer, or any client system using software for the NFS protocol.

In MVS, a file is called a data set. The NFS allows client users to mount conventional MVS data sets from their workstations. It presents the information to them in the form of a UNIX (or AIX) or DOS file, though the information is actually stored on an MVS-owned DASD.

The files for an operating system are organized into a file system. The UNIX and DOS environments use a file system that is a hierarchy of directories. Conventional MVS, in contrast to z/OS UNIX, uses a non-hierarchical file system in which groups of data sets are referred to by specifying a high-level qualifier (HLQ).

The MVS HLQ can include the first (leftmost) qualifier of data sets, or the first and second qualifiers, or the first, second, and third qualifiers, and so on. For example, SMITH is the HLQ for the files named SMITH.TEST.DATA and SMITH.PROJ7.SCHED, while SMITH.TEST is the HLQ of SMITH.TEST.DATA and SMITH.TEST.DOCS.

## Mounting MVS data sets onto a client mount point

To access an MVS file system from your client, client users use the **mount** command to create a temporary link (until unmounted) between specific MVS data sets and your UNIX directory (preferably empty) or an unused logical drive on their workstations. The empty UNIX directory or logical drive is called a *mount point*.

Client users use an MVS HLQ in the **mount** command to specify which MVS data sets to mount at a mount point. The MVS data sets beginning with the specified HLQ appears as files under the mount point.

Client users can also perform a mount using a fully qualified data set name or an alias to a user catalog, but not the catalog name itself. Only Integrated Catalog Facility (ICF) cataloged data sets are supported by the z/OS NFS server. Tape data sets and generation data sets are not supported.

Some client platforms support TCP in addition to UDP. Users can choose either TCP or UDP to access the server. The default protocol option depends on the NFS client platform.

**Note:** When directly mounting on a fully qualified data set name, the server must return the mount size as part of getting the attributes for the mount. This can slow down the completion of the **mount** command.

## Crossing file systems–NFS server

Crossing file systems means the NFS client can also potentially be a server, and remote and local mounted file systems can be freely mixed. This leads to some problems when a client travels down the directory tree of a remote file system and reaches the mount point on the server for another remote file system. Allowing the server to follow the second remote mount would require loop detection, server lookup, and user revalidation. When a client does a lookup on a directory on which the server has mounted a file system, the client sees the underlying directory instead of the mounted directory.

The NFS server does not support crossing file systems. For example, if a server has a file system called */usr* and mounts another file system on */usr/src*, a client can also mount */usr*, but the server will not see the mounted version of */usr/src*. A client could perform remote mounts that match the server's mount points to maintain the server's view. In this example, the client would also have to mount */usr/src* in addition to */usr*, even if the mounts are from the same server.

## Creating conventional MVS data sets

Client users can create MVS data sets from a client system using the NFS. The default data set creation attributes specified by the system administrator are used to create MVS data sets, unless the user overrides them. These attributes determine how the MVS data sets are structured and where they are stored. Client users can override the default data set creation and processing attributes for a mount point when issuing the **mount** command. In addition, you can override these attributes at file creation time.

## Data set serialization and sharing

The z/OS NFS server handles data set serialization and sharing differently, depending on the type of data set:

Physical sequential	The server insures the physical sequential data set read/write integrity by SVC 99 dynamic allocation with exclusive option whenever a physical sequential data set is opened for output. Otherwise, it allocates with share option.
Virtual Storage Access Method (VSAM)	The server dynamically allocates a VSAM data set with share option and allows the VSAM access method to manage data sharing using the <b>shareoptions</b> specified during data set definition.
Partitioned data set extended (PDSE)	The server dynamically allocates a PDSE data set with share option and allows the PDSE functions to manage the serialization of the PDSE data set and its members.
Partitioned data set (PDS)	For read and write, the z/OS NFS server issues ENQ SHR on QNAME=SYSDSN and RNAME=dataset_name (through an SVC 99). For write, the server issues an exclusive ENQ against QNAME=SPFEDIT and RNAME=dataset_name.member_name, in addition to the serialization of resources by SVC 99. For all MVS users who are allocating their data set with exclusive status, this provides write protection. It only provides read integrity for ISPF users.

---

## Server control files

These special files are used by the MVS system administrator to control the z/OS NFS server:

- Attributes data set
- Exports data set
- Mount handle data sets
- Log data sets
- Checklist data set

## Attributes data set

The attributes data set contains the settings for the z/OS NFS server. There are three types of attributes stored in this data set:

Data set creation attributes	Used to define the structure of MVS data sets when creating a file (for conventional MVS data sets only)
File processing attributes	Used to control how files are accessed by the client
Site attributes	Used to control z/OS NFS server resources

The system administrator changes the default settings by editing the attributes data set and restarting the server. Client users can override the data set creation and file processing attributes at the command line. For conventional MVS data sets, the client user can specify the data set creation attributes when mounting, creating, or accessing files. The client user can override the file processing attributes when mounting, creating, or accessing files. However, some file processing attributes can only be overridden on a *mount point* basis.

**Note:** Many of the attributes are valid for conventional MVS files only, and not for z/OS UNIX files. “Attributes used for z/OS UNIX System Services file access” on page 75 gives a complete list of attributes that are valid for z/OS UNIX.

## Exports data set

The exports data set can control which client users can mount which MVS data sets. The entries in the exports data set specify which MVS high-level qualifiers or HFS directories can be mounted. The system administrator can use this data set to limit mounts to accredited clients only. It also controls which client users can mount all or part of the z/OS UNIX file system, based on the client machine’s specified Internet Protocol (IP) address. To use the exports data set, the **security** site attribute must be set to either **safexp** or **exports** by the MVS system administrator.

## Mount handle data set

The z/OS NFS server maintains a list of the active *mount points* in a pair of files called the mount handle data sets on MVS. The two data sets are used alternately to automatically reestablish the client *mount points* when the server is started. If the file system is not available, the mount point is not reestablished and the mount failure is recorded in the LOG data set.

The z/OS NFS server does the cleanup activity during z/OS NFS server shutdown and daily at the cleanup time specified by the **restimeout** site attribute.

During cleanup time, the z/OS NFS server reads the list and checks all mount points against the retention period specified in the **restimeout** site attribute. If your *mount points* are idle longer than the retention period specified in the **restimeout** site attribute, they are removed. Only the active *mount points* are reconnected.

If a mount handle is removed by the cleanup activity, the client user might receive the “Stale NFS File Handle” message or some other appropriate message. If so, all the client user needs to do is unmount the stale *mount point* and mount it again.

## Log data set

The *log data sets* store the messages for the z/OS NFS start-up procedures. This log can be used to identify the users correctable errors or the users problem errors. There are two logs that this information is stored in; the primary log and the secondary log. The primary log is used at start-up until it is filled and then overflows into the secondary log. When the secondary log is full, the primary log will then be overwritten with new error messages. The number of log records is dependent on the number of transactions that the server can handle.

## Checklist data set

The checklist data set contains entries for files or directories that are to be exempt from System Authorization Facility (SAF) checking even though SAF or SAFEXP is specified as the security options. This file is only used when SAF or SAFEXP is specified for the particular data type and the **checklist** option is specified as a site attribute. The entries specified here must match a subsequent mount point, or be the parent of a subsequent mount point, to allow SAF checking to be bypassed for everything underneath that mount point. If the entry does not match a subsequent mount point and it is not a parent of any subsequent mount point then it has no effect.

---

## Supported clients for the z/OS NFS server

The following IBM systems are available for NFS client:

- z/OS NFS
- IBM RS/6000® AIX version 4.2.0

NFS client software for other IBM platforms is available from other vendors. You can also access the z/OS NFS server from non-IBM clients that use the NFS version 2 protocol, including:

- DEC stations running DEC ULTRIX version 4.4
- HP 9000 workstations running HP/UX version 10.20
- Sun PC-NFS version 5

You can also access the z/OS NFS server from non-IBM clients that use the NFS version 3 protocol, including:

- Sun workstations running SunOS or Sun Solaris versions 2.5.3
- AIX 4.2.0

**Note:** This information reflects supported clients as of September 1998. For current information on supported clients, contact IBM.

---

## Supported servers for the z/OS NFS client

The z/OS NFS client supports all servers that implement the server portion of the Sun NFS Version 2 or Version 3 protocols.

A **mount** parameter **vers(x)**, where x is either 2 or 3, is provided to make the z/OS NFS client communicate with the server at the specified protocol level. The z/OS NFS client also communicates at the highest protocol level that is supported by the server if no level is specified.

If **no version** is specified and if the server:

- Only supports NFS version 2 protocol, then the z/OS NFS client will use NFS version 2 protocol to communicate
- Supports both the NFS version 2 and 3 protocols, then z/OS NFS client will use NFS version 3 protocol to communicate

If **vers(2)** is specified, then use NFS version 2 protocol to communicate with the server.

If **vers(3)** is specified, then use NFS version 3 protocol to communicate with the server. z/OS NFS client fails the **mount** command if the server does not support NFS version 3 protocol.

---

## WebNFS support

The z/OS NFS server supports the WebNFS protocol. WebNFS specification extends the semantics of NFS versions 2 and 3 protocols to allow clients to obtain file handles without the mount protocols. The z/OS NFS server supports the public filehandle and multi-component **lookup** features as well as other additional requirements as described in RFC 2055. A new keyword, **public**, is added for the system administrator to specify the public paths that the public file handle can access. A public path for conventional MVS data and a public path for HFS data can both be specified. When a **lookup** request comes in from an NFS client and an absolute path name is specified, it will be matched with the public paths to determine which public path it is trying to reference. If a relative path is specified and both HFS and MVS public paths are defined then the **lookup** request will be processed relative to the HFS public path.

The following are restrictions for the WebNFS support provided by the z/OS NFS server in this release.

**Export Spanning Pathnames** - **lookup** requests, which reference files or directories outside of the exported public path, will result in an error condition.

**Symbolic Links** - A symbolic link embedded in a multi-component pathname **lookup** request will result in an error condition. However, if the final component is a symbolic link, the server will return the filehandle of the symbolic link and let the client evaluate it. External links, which are special cases of symbolic links, will be handled similarly.

**Native Path** - Only canonical pathnames will be supported.

---

## NFS version 3 and TCP protocol

The information for NFS version 3 protocol and **proto=tcp** can be found on the *mount* man page on a UNIX client. The NFS client automatically selects the option unless the end-user overrides the option. For example, you can enter this command:

```
unix$ mount -o vers=2,proto=udp mvshost1:smith /mnt
```

The above example shows the preference of NFS version 2 with **udp** protocol, even though the client platform can handle the NFS version 3 and **tcp** protocol.

Users can issue the `rpcinfo -p <hostname>` to show all the RPC programs available on the server. Table 3 illustrates this example.

```
$ rpcinfo -p mvshost1
```

Table 3. View of NFS server capability

program	vers	proto	port	service
100000	2	udp	111	portmapper
100000	2	tcp	111	portmapper
100003	2	udp	2049	nfs

Table 3. View of NFS server capability (continued)

program	vers	proto	port	service
<u>100003</u>	<u>3</u>	<u>udp</u>	<u>2049</u>	<u>nfs</u>
100059	2	udp	2012	
100044	1	udp	2013	
100005	1	udp	2014	mountd
100005	3	udp	2015	mountd
150001	1	udp	2016	pcnfsd
150001	2	udp	2017	pcnfsd
<u>100044</u>	<u>1</u>	<u>tcp</u>	<u>2010</u>	
<u>100059</u>	<u>2</u>	<u>tcp</u>	<u>2011</u>	
<u>100003</u>	<u>2</u>	<u>tcp</u>	<u>2049</u>	<u>nfs</u>
<u>100003</u>	<u>3</u>	<u>tcp</u>	<u>2049</u>	<u>nfs</u>

**Note:** New information is represented in underlined text.

---

## Native ASCII support

The z/OS NFS client and server support applications running on z/OS V1R2 (and higher) in a "Native ASCII" environment. Applications can operate on files in either EBCDIC or ASCII format as well as other data formats defined with a Coded Character Set Identifier (CCSID). Native ASCII support is provided with a mechanism called file tagging where the file is defined with a tag to identify the CCSID to use for data conversion. File tagging is defined in the appropriate z/OS UNIX System Services (USS) publications. The z/OS NFS client and server provide the necessary support to provide data conversion between different CCSIDs specified for the client and server. The z/OS NFS client *cln\_ccsid* and *srv\_ccsid* parameters will also be supported by the z/OS NFS server to identify the CCSID to be used in the data conversion. See "Processing attributes" on page 78 for more information about the *cln\_ccsid* and *srv\_ccsid* parameters.



---

## Chapter 2. Using conventional MVS data sets

This chapter explains what you need to know to use conventional MVS™ data sets on a client workstation. This chapter discusses the following topics:

- Special MVS considerations
- Reading and writing MVS data sets
- Accessing MVS data sets
- Mapping between the workstation and MVS file systems

---

### Special MVS considerations

In addition to mapping between the workstation and MVS file systems, there are other ways in which the z/OS NFS server might be different from non-MVS NFS servers. These differences include:

- Selecting an MVS data storage format
- File size determination and time stamps
- Ownership and permissions
- Reading and writing Files
  - Random access to files
  - Cached data writing
- Statelessness
- Case sensitivity - maplower, nomaplower
- Selecting text or binary processing modes - text, binary
  - Binary processing mode
  - Text processing mode
- Number representation
- Accessing migrated files - retrieve, noretrieve; wait, nowait
- Obtaining attributes of migrated data sets without recall
- File handle refresh
- File extension mapping

### Selecting an MVS data storage format

The files you create with the z/OS NFS server are contained in MVS data sets. These MVS data sets are record-oriented and can be sequential, direct, VSAM, partitioned, and so forth. These MVS data sets are variable or fixed in record length. UNIX, OS/2, and DOS files, however, are byte-oriented and typically written or read at certain offsets in the file.

You can map non-MVS files to most types of MVS data set organizations. However, how the time stamps and file size value are handled depends on the type of MVS data set used, and the file size processing can affect performance. See “Appendix C. Handling of the file size value” on page 143.

Direct reads with record format `recfm(fbs)` or `recfm(f)` can be fast. In some cases, the z/OS NFS server can determine the physical block addresses from the record offsets. The MVS sequential file organization with `recfm(fbs)` or `recfm(f)` on DASD allows for efficient updating or reading at any offset in the file. Other supported MVS access methods (for example, VSAM) can be used if required by a given application but, in general, the sequential file organization is the best choice for files that are used mainly by UNIX clients.

## File size determination and time stamps

How the z/OS NFS server handles the file size value and time stamps depends on the type of MVS data set used and the attributes used to access the data set. See “Appendix C. Handling of the file size value” on page 143 and “Appendix D. Handling of the time stamps” on page 147.

## Ownership and permissions

The UNIX UID and GID file attributes are reset to their default state after a restart of the z/OS NFS server or an unmount of the file system. In some cases, this requires that a superuser on the client workstation reissue a **chmod** command to reset the UID and GID. This command can be included in the same script used to mount the file system.

The permissions checking done by Resource Access Control Facility (RACF®), or an equivalent security package, is transparent to you. Access to a data set is granted, provided that the server’s exports list, the MVS security subsystem, and the customized installation security exit allow access to the data set. Which of these security systems are active depends on the security settings used at your installation. The UNIX file modes or permission bits are ignored by z/OS NFS server and authorization is done with the RACF or equivalent security package.

## Statelessness

The z/OS NFS server strives to be as stateless as possible; that is, it tries to work correctly without maintaining any state information about any of its clients. However, a failure of the server causes cached writes to be lost, some attributes to be reset, and file handles to become stale, or not valid.

## Reading and writing files

After the z/OS NFS server is started and you have mounted the MVS data set, you can use regular data access or creation commands from your workstation to access files that reside on MVS.

For example, suppose you accessed an MVS file named *prefix.file3* mounted on the local directory */mnt*. This is how you could use the UNIX **cat** command (or a similar command) to display the file:

```
$ cat /mnt/file3
```

Suppose you accessed an MVS file named *prefix.file12* mounted on the local directory */mnt*. This is how you could use the UNIX **vi** command (or a similar command) to edit the file:

```
$ vi /mnt/file12
```

Writing a file on MVS is straightforward. If the file already exists, the file’s existing attributes are used; they are not modified during the write operation. For the priorities of attributes, see “Overriding data set creation attributes” on page 21.

### Random access to files

If your application accesses the files at random offsets, there is a performance implication.

In the NFS environment, a file is represented as a byte stream. That byte stream is accessible for reading and writing at any byte offset for any byte length. In the MVS environment, a file is represented as a collection of records. The record, rather than a single byte, is the smallest object that can be processed. Therefore, the z/OS NFS server has to convert the byte stream operations from NFS clients into standard access method operations on MVS.

To convert byte stream operations to MVS access method operations, the server has to determine which record in the MVS file contains the offset specified in the NFS read or write request. To determine this, the server reads, mapping byte offsets to records, from the last known location in the file until the record containing the requested byte offset is located.

For example, suppose a file on MVS contains 10,000 variable-length records with a maximum length of 80 bytes for any record. Suppose the first NFS request received tells the server to read 4,000 bytes starting at offset 10,000 bytes. Because the file has not been opened yet, the server would open the file and start reading at the first record, searching for the record that contains offset 10,000. Once it found the record, the server would process the request, which might involve reading more records to find enough bytes to satisfy the request.

Another complication involved in mapping byte offsets to records is the processing defined by the user to apply to a file. For example, if you specify text mode processing with line terminators, the perceived offset into a file from a given client changes.

### Cached data writing

The z/OS NFS might cache writes if out-of-sequence data packets are received or if a block of data is partially filled. If the Network File System is processing in the binary data, the writes will remain cached until one of the following occurs: the timeout for the request has been reached or the number of cached packets exceeds the number specified in ***cachwindow()***. If the NFS is processing text data, the writes remain cached until a timeout occurs. The missing data is padded with binary zeroes and record delimiters so that cached writes for text processing are written in the MVS data set on DASD at the location specified in each cached data packet. In the case of cached data packets for binary processing, only binary zeroes will be used to pad the missing data written at the specified location on DASD. See the Table 4 table below.

**Attention:** For NFS version 3 ***commit*** procedure, z/OS server will only support committing the cached data when the data set is timed out.

Table 4. Breakdown of text and binary writes

Description	Binary	Text
Data is flushed to DASD when the	Number of cached packets exceeds the amount specified in <b><i>cachwindow()</i></b> , or the file times out.	File times out. If the number of packet exceeds the amount specified in <b><i>cachwindow()</i></b> , all new out of sequence packets will be dropped.
Padding	Binary zeros	Binary zeros
Record delimiters	There are no record delimiters. Therefore, there is no attempt to add end of line characters.	There can be record delimiters. Therefore, an end of line character is added to the end of the record.

## Case sensitivity-maplower, nomaplower

If the processing attribute **maplower** is specified, the MVS file name is mapped to the lower case when returned to the client. If the processing attribute **nomaplower** is specified in the attributes, all entries in the exports data set are case-sensitive. Therefore the client mount request must specify the MVS qualifier with the correct case to successfully match the exports data set entry.

## Selecting text or binary processing modes-text, binary

You can specify either text or binary processing mode when you access files. This processing mode does not describe the type of data in the original file, but rather, it specifies whether to convert between ASCII and EBCDIC when sending file contents between the MVS host and the client workstation. See "Mount command syntax and examples" on page 57 for additional information on text and binary processing of files using the z/OS NFS client.

### Binary processing mode

The binary processing mode specifies to send and receive file contents between the MVS host and the client in binary form, avoiding the ASCII/EBCDIC conversion required in text mode. This is faster than text mode. However, users on MVS cannot read the file, because the contents are not in EBCDIC. Therefore, use the binary processing mode to create or access a file only if the file is not intended to be shared with users on the MVS host, or the file content is binary.

When fixed-length records are written in binary mode, the server pads the last record of the file with null characters if the last record is less than the fixed record length. These padding bytes are counted in the file size.

### Text processing mode

With the text processing mode, when data is read, record boundaries are converted to (or from) workstation line terminators such as '\n' or '\r\n', and data representation is changed between EBCDIC and ASCII.

**Selecting how blanks are handled-blankstrip, noblankstrip:** When fixed-length records are written in text mode, records are padded with blanks if the record length is larger than a line, and if the **blankstrip** processing attribute is enabled. (When sending data from MVS, blankstrip strips trailing blanks. When sending data to MVS, blankstrip pads the records with blanks). In text mode processing, data representation is changed from ASCII to EBCDIC, and line terminators are converted to record boundaries. All data is converted according to the active translation table. Therefore, if the data set contains a mixture of characters and binary data, binary data is converted as well. In text mode, be careful not to mix your text data (characters) with binary data.

If you are writing data to a fixed-length MVS file in text mode with blank stripping enabled, and the data contains blanks at the end of the line, an I/O error occurs. This is because the server is not able to return the blanks to the client when the file is read back.

If you save a fixed-length MVS file in text mode with one or more lines exceeding the maximum record length, an I/O error occurs. For example, suppose an MVS file has fixed-length records of 80 bytes. After you edit the file using the vi editor on your workstation, one of the file's records is 83 bytes long (exceeding the fixed length by 3 bytes). When you save the file back to the server, the MVS file is either partially or totally destroyed, and the "I/O Error" message appears on your screen.

While you are still in the editing session, save the edited file in an alternate local file. After you correct the local file so that no line exceeds 80 bytes, save it back into the MVS file.

If you get an error message when trying to create or access an MVS file, see "Appendix A. Network File System messages" on page 93 for further explanation of the message.

**Using crlf or if line terminators:** For a DOS or OS/2 client, use "crlf" as the line terminator sequence when accessing data in text processing mode. For an AIX or UNIX client, use 'lf' as a line terminator when using text processing mode.

## Number representation

The **text** processing mode does not change the number representation format between the host and client. When you choose **text** as the processing mode, the NFS converts characters between ASCII format and EBCDIC format, and processes line terminator, but no other translation of user data occurs.

When you select binary as the processing mode, NFS stores your data unchanged. Therefore, regardless of the processing mode you choose, you cannot change numbers from one client workstation's format to another client workstation's format.

## Accessing migrated files - retrieve, noretrieve; wait, nowait

Sometimes files on MVS are migrated to another storage level, such as a space-saving format on DASD or tape. If your file has been migrated and you try to access it, it might take awhile for it to be recovered back into primary storage. The **retrieve** and **noretrieve** processing attributes control what happens when you try to access a migrated file.

There are three ways that the retrieve or noretrieve option is controlled.

1. Using the Default Retrieve Attribute

You can use the default retrieve processing attribute by not entering **retrieve** or **noretrieve** in your **mount** command or file access command.

2. Specifying retrieve with the **mount** command

You can issue the **mount** command, specifying **retrieve** or **noretrieve**. The attributes specified in the **mount** command override the attributes in the default attribute data set.

In this example, migrated files under the mount point are not retrieved. However, you can access files under the mount point which are not migrated.

```
$ mount mvshost1:"smith,noretrieve" /u/smith/mnt
```

Conversely, the next command causes the migrated files under the mount point to be retrieved when accessing the files.

```
$ mount mvshost1:"smith,retrieve" /u/smith/mnt
```

3. Specifying retrieve with a file access command

You can issue a file access command with the attribute **retrieve** or **noretrieve** specified. The attributes specification in the file access command overrides the attributes in the **mount** command and the server default attributes.

This command causes all files under the mount point /u/smith/mnt to be retrieved if they are migrated:

```
$ ls -l "/u/smith/mnt,retrieve"
```

This command, however, does not cause migrated files under the mount point /u/smith/mnt to be retrieved:

```
$ ls -l "/u/smith/mnt,noretrieve"
```

## Accessing migrated system-managed data sets

DFSMS/MVS Version 1 Release 3 allows data set attribute accessibility for SMS-managed data sets, without having to recall the data set if the data set is migrated under DFSMS/MVS V1R3. Supported data set types are SMS-managed PS, VSAM ESDS, VSAM KSDS, VSAM RRDS, PDS, and PDSE. Migrated PDS/PDSE members are not supported.

The z/OS NFS server is able to obtain the attributes of a supported SMS-managed migrated data set without recalling the data set. Attributes such as the time stamp and file size are saved to DASD. Subsequent file size requests do not cause a recall of the supported SMS-managed migrated data set, thus improving performance. However, when the data set is modified outside the server by a non-NFS application (for example, by the TSO editor) before it was migrated, the stored file size could be incorrect. When the data set is accessed again by the server, a recall must be done to determine the correct file size.

When a request is made to remove any of the supported SMS-managed migrated data sets, the data set will be deleted without recall. For PDS and PDSE migrated data sets, the data set will be recalled in order to read its member information.

## File handle refresh

File handles of mounted objects (directories or file systems) are saved on DASD in a mount handle data set and are automatically established again when the server restarts. However, file handles for the files within a mounted object are kept in virtual storage (memory) and they are lost if the server restarts. This results in stale file handles, and the clients must request a new file handle by redoing the **lookup** on the file.

---

## Mapping between the workstation and MVS file systems

In MVS, a file is called a data set, and the two terms are used interchangeably in this book. The z/OS NFS server presents information to you in the form of a UNIX (or AIX), OS/2, or DOS file, though the information is actually stored on MVS-owned DASD in the form of an MVS data set.

The files for a computer system are organized into a file system. The UNIX, OS/2, and DOS environments use a file system which is a hierarchy of directories. MVS, however, uses a non-hierarchical file system in which groups of data sets are referred to by specifying a high-level qualifier.

The MVS high-level qualifier can include the first (leftmost) qualifier of data sets, or the first and second qualifiers, or the first, second, and third qualifiers, and so on. For example, SMITH is the high-level qualifier for the files named SMITH.TEST.DATA and SMITH.PROJ7.SCHED, while SMITH.TEST is the high-level qualifier of SMITH.TEST.DATA and SMITH.TEST.DOCS.

## File name extension mapping

File extension mapping allows users to access members of PDS/PDSE data sets on the MVS host which are mapped from client machine files that contain file extensions. Each PDS/PDSE data set on the host can only be mapped with one unique file extension. For example: IBMUSER.TEXT(M1), IBMUSER.TEXT(M2) will map to m1.txt, m2.txt under directory ibmuser.text on the client machine. This capability allows client machine tools such as editors and compilers, to process host files remotely without modification. There are site and processing attributes that are associated with the file extension mapping. The client user can specify the **fileextmap** attribute to turn the file extension mapping on and the **nofileextmap** attribute to turn the file extension mapping off. The **sidefile(dsname)** attribute allows the client user to specify a side file name that is different than the default side file name during the mount operation. For example you can enter this command:

```
[C:\] mount z: mvshost1:"user1.pds,sidefile(hlq.nfs.mapping)"
```

The side file specified at the **mount** command will be searched first followed by the default side file. The system administrator can specify the maximum space available for side files using the new site attribute **sfmax (n)**. A sample mapping side file is provided as GFSAPMAP in the NFSSAMP library. See *z/OS Network File System Customization and Operation* for more information.

## Mounting MVS data sets onto a client mount point

To access an MVS file system from the client, you use the **mount** command to create a temporary link between specific MVS data sets and a UNIX directory (preferably empty) or an unused logical drive. The UNIX directory or drive is called a mount point.

You use an MVS high-level qualifier in the **mount** command to specify which MVS data sets to mount onto a mount point. The MVS data sets beginning with the specified high-level qualifier appear as files under the mount point. See Figure 2 on page 16.

You can also perform a mount using a fully qualified data set name or an alias to a user catalog, but not the catalog name itself. Only Integrated Catalog Facility (ICF) cataloged data sets are supported by the z/OS NFS server, and tape data sets are not supported.

Data set organizations supported include:

- Physical sequential (PS)
- Direct access (DA)
- Partitioned data sets (PDS)
- Partitioned data sets extended (PDSE)
- VSAM KSDS
- VSAM ESDS
- VSAM RRDS
- HFS
- SAM extended format data sets

Both SMS-managed and non-SMS-managed data sets are supported.

- Multi-volume data sets are not supported except for extended format data sets.
- Generation data sets are also not supported.

### Notes:

1. The filesize for the MVS conventional data set as a directory has a dummy size with a value of 8192.
2. For NFS Version 3 CREATE procedure, z/OS server does not harden the exclusive create verify token to disk.

### Variants of the mount command

The name of the command that performs the mount operation varies for different client implementations. For example, the SUN PC-NFS implementation uses the **net use** command while most UNIX implementations use the **mount** command.

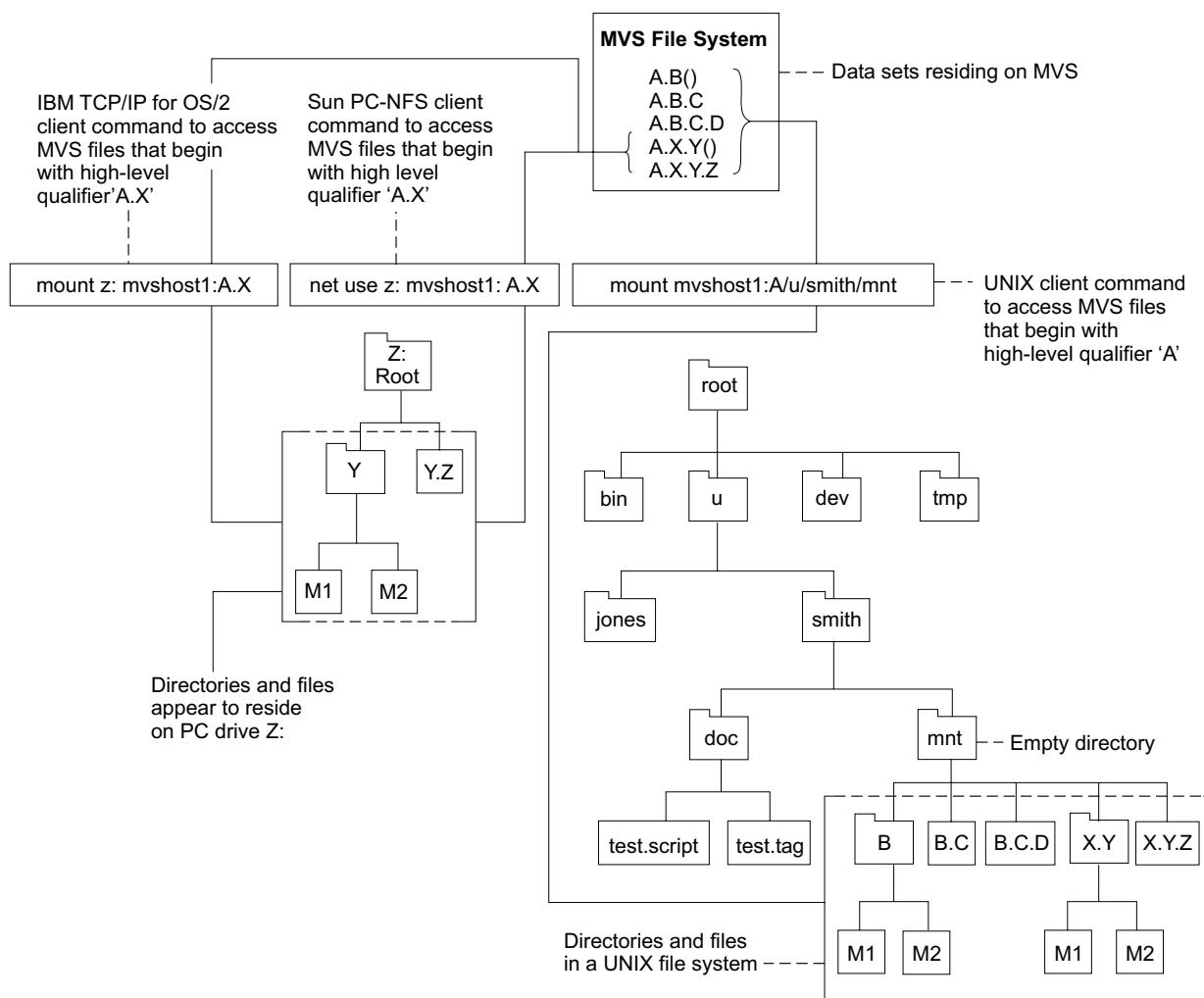


Figure 2. Examples of mounting MVS data sets on OS/2, DOS, and UNIX clients. The OS/2 and DOS clients are mounting the MVS data sets which begin with the high-level qualifier "A.X". The UNIX client is mounting the MVS data sets which begin with the more inclusive high-level qualifier "A". The parentheses indicate a PDSE containing the members M1 and M2. The PDSEs A.B() and A.X.Y() appear as directories, and their members appear as files within those directories.

### Using a PDS or PDSE as a directory

If the data sets specified include partitioned data sets, a second level of hierarchy is shown. This allows you to define one level of directories under the mount point. Thus, you can issue **mkdir** to create a directory (stored as a PDS or PDSE) and then create files (stored as members of a PDS or PDSE) within that directory.



This use of a PDSE is shown in Figure 2 on page 16, which illustrates the mapping of file names between client file systems and the MVS file system resulting from a **mount** command.

## Using multiple mount points

You can arrange groups of data sets into several UNIX mount directories (or PC mount drives) by using MVS naming conventions to mount specific data sets at each mount point. For example, you could mount user1.project1 to get all data sets beginning with "user1.project1" mounted at one point in the local file system, and you could mount user1.project2 at another point. This would create the effect of two distinct directories (or drives), one containing the user1.project1.\* data sets, and the other containing the user1.project2.\* data sets.

## Data set serialization and sharing

The z/OS NFS server handles data set serialization and sharing differently depending on the type of data set:

**Physical sequential** - The server insures the read/write integrity of a physical sequential data set by SVC 99 dynamic allocation with exclusive option whenever a physical sequential data set is opened for output; otherwise it is allocated with share option.

**VSAM data set** - The server dynamically allocates a VSAM data set with share option and allows the VSAM access method to manage data sharing via the SHAREOPTIONS specified during data set definition.

**PDSE data set** - The server dynamically allocates a PDSE data set with share options and allows the PDSE functions to manage the integrity of the PDSE data set and its members.

**PDS data set** - The server dynamically allocates a PDS data set with share option and surrounds the PDS and its members with exclusive ENQs against the QNAME=SPFEDIT and RNAME=data set name. This does not protect the PDS from other MVS users who are attempting to access the PDS without performing ENQ against SPFEDIT similar to the z/OS NFS server.

## NFS protocol

Table 5 illustrates that the NFS procedures are not all supported for conventional MVS data sets.

*Table 5. NFS procedures*

Procedure	Version 2 Protocol	Version 3 Protocol
link	no	no
mknod	N/A	no
readlink	no	no
readdirplus	N/A	no
setattr	yes	yes
statfs	yes	N/A
symlink	no	no

**Note:** Setattr only supports filesize=0 truncation and UNIX permission is set to 777.

## File system size

The server generates values for the UNIX attributes of file system block size and file system size in blocks, because they have no meaning in the MVS context, except for HFS files. Table 6, Table 7, and Table 8 illustrate the MVS values that the server generates.

Table 6. File system values to get dynamic file system information (fsstat)

Value	Description	Conventional MVS Value
tbytes	Total size, in bytes, of the file system	10000000000
fbytes	Amount of free space, in bytes, in the file system	8000000000
abytes	Amount of space, in bytes, available to the user identified by the authentication in the RPC	80000000
tfiles	Total number of file slots in the file system	200000
ffiles	Number of free file slots in the file system	20000
afiles	Number of free file slots that is available to the user corresponding to the authentication information in the RPC	2000
invarsec	Number in seconds for which the file system is not expected to change	0

Table 7. File system values to get static file system information (fsinfo)

Value	Description	Conventional MVS Value
rtmax	Maximum number in bytes for the read request supported by the server	65536 (64KB)
rtpref	Preferred size of the read request	32768 (32KB)
rtmult	Suggested multiple for the size read request	4096
wtmax	Maximum size of a write request supported by the server	65536 (64KB)
wtpref	Preferred size of the write request	32768 (32KB)
wtmult	Suggested multiple for the size of a write request	4096
dtpref	Preferred size of the readdir request	8192
maxfilesize	Maximum size of a file on the system	2 ** 63 - 1
time_delta	File time using setattr	(0,1000000)
Properties:	FSF_LINK	1
	FSF_SYMLINK	0
	FSF_HOMOGENEOUS	1
	FSF_CANSETTIME	1

Table 8. File system values to retrieve POSIX information (pathconf)

Value	Description	Conventional MVS Value
linkmax	Maximum number of hard links	1
name_max	Maximum length of a component file name (file name + attributes)	255
no_trunc	Server will reject any name that is longer than the name_max	True

*Table 8. File system values to retrieve POSIX information (pathconf) (continued)*

<b>Value</b>	<b>Description</b>	<b>Conventional MVS Value</b>
chown_restricted	Change either the owner or the group associated with the data set	True
case_insensitive	Server does not distinguish the case when interpreting file names	True
case_preserving	If True, the server file system will preserve the case of a name during a create, mkdir, mknod, symlink, rename, or link	False



---

## Chapter 3. Creating conventional MVS data sets

This chapter explains how to create the various types of data sets (files) that are supported by the z/OS NFS server.

The examples shown are for the AIX RS/6000 platform. Any examples for other platforms are so indicated. For PC platforms see “Chapter 7. Commands and examples for Sun PC-NFS clients” on page 49.

---

### Overriding data set creation attributes

When you create an MVS file, default file creation attributes are applied, unless you override them. The attributes are passed to the MVS host.

Data set creation attributes are controlled in the following ways, in increasing order of priority:

- Default server data set creation attributes
- Default installation data set creation attributes, specified by the system administrator in the attributes data set
- DFSMS data class attributes
- Data set creation attributes specified in the **mount** (or **net use**) command
- Data set creation attributes specified in the **mkdir**, **vi** (edit), or **cp** (copy) commands (highest priority)

**Note:** The z/OS NFS server does not support these data class attributes:

- Retention period/Expiration date
- Number of volumes
- VSAM imbed index option
- VSAM replicate index option
- CI size of data component
- Percentage of CI or CA free space

---

### Preparing to create an MVS file

When creating an MVS file, you should know whether to process the file in text or binary mode (see “Selecting text or binary processing modes-text, binary” on page 12) and what type of file to create.

The z/OS NFS server supports these types of files:

- Physical sequential (PS)
- Direct access (DA)
- Partitioned data sets (PDS)
- Partitioned data sets extended (PDSE)
- VSAM KSDS
- VSAM ESDS
- VSAM RRDS
- SAM extended format data sets

Keyed access to files and GDG are not supported.

### Naming MVS files

The NFS uses comma (,) as a delimiter to a list of file attributes. Do not use (,) as a special character in file name. For example, you can enter this command:

```
$ vi "/u/smith/new,text"
```

This indicates to NFS that a file called `new` is being edited in the attribute text mode, not file `new,text`.

When naming conventional MVS files, you must follow the MVS file naming conventions, as described in *z/OS DFSMS: Using Data Sets*.

For information about the z/OS UNIX System Services naming conventions see “Chapter 4. Using z/OS UNIX System Services files” on page 29.

An MVS file name (or data set name) can consist of one or several simple names joined so that each represents a level of qualification. For example, the MVS file name `DEPT58.SMITH.DATA3` is composed of three qualifiers.

The following characteristics apply to the MVS file name:

- Each qualifier consists of 1 to 8 alphanumeric characters, national characters (@, #, \$), or a hyphen (-).
- Each qualifier must start with an alphabetical or national character.
- The period (.) separates simple names from each other.
- Including all simple names and periods, the length of the MVS file name must not exceed 44 characters.
- PDS and PDSE member names can be up to 8 characters long.

For information about the MVS file system, see “Mapping between the workstation and MVS file systems” on page 14.

### Restrictions using alias names for MVS files

For non-VSAM files, alias names can be used interchangeably with the true file name except on remove (**rm** and **rmdir**) and rename (**mv**) requests. Renaming or removing an alias name results in an I/O error. This is due to an MVS restriction.

If the true name of an MVS file is renamed or removed and alias names have been defined for the file, MVS deletes the alias names during execution of the rename or remove request.

---

## Creating physical sequential files

When creating a physical sequential (PS) file, specify the **dsorg(ps)** attribute (if it is not the default already) with the **mount** command or a file creation command, such as the **vi** UNIX (or AIX) command.

1. Create a local directory on your client to be used as a mount point. For example (with UNIX), enter this command:

```
$ mkdir /u/smith/mnt
```

2. Mount the MVS file system. For example, suppose your host is `mvshost1`, and you want to issue a mount on the high-level qualifier `smith`. You can enter this command:

```
# mount mvshost1:"smith,dsorg(ps)" /u/smith/mnt
```

If you do not specify any other attributes, the MVS site defaults are used. You can use the **showattr** command to display the site defaults.

3. You can use the **vi** UNIX command to create the new file:

```
$ vi /u/smith/mnt/new
```

When you save the file using **vi**, you have just created a new MVS PS file named SMITH.NEW.

You can get the same results by specifying **dsorg(ps)** in the file creation command rather than in the **mount** command:

```
# mount mvshost1:smith /u/smith/mnt
$ vi "/u/smith/mnt/new,dsorg(ps)"
```

---

## Creating direct access files

When creating a direct access (DA) file, specify the **dsorg(da)** attribute (if it is not the default already) with the **mount** command or a file creation command (such as the **vi** UNIX command).

1. Create a local directory on your client to be used as a mount point. For example (with UNIX), enter this command:

```
$ mkdir /u/smith/mnt
```

2. Mount the MVS file system. For example, suppose your host is mvshost1, and you want to issue a mount on the high-level qualifier smith. You can enter this command:

```
# mount mvshost1:"smith,dsorg(da)" /u/smith/mnt
```

If you do not specify any other attributes, the multiple virtual system (MVS) site defaults are used. You can use the **showattr** command to display the site defaults.

3. Next, you can use the **vi** UNIX command to actually create the new file:

```
$ vi /u/smith/mnt/new
```

You have just created a new MVS DA file named SMITH.NEW.

You can get the same results by specifying **dsorg(da)** in the file creation command, rather than in the **mount** command:

```
# mount mvshost1:smith /u/smith/mnt
$ vi "/u/smith/mnt/new,dsorg(da)"
```

---

## Creating PDSs and PDSEs

Partitioned data sets (PDSs) and partitioned data sets extended (PDSEs) can be used as directories, and their members are files within those directories. An illustration of the use of PDSs to act as directories is shown in Figure 2 on page 16. For general information on PDSs and PDSEs, see *z/OS DFSMS: Using Data Sets*.

You cannot create new directories within a PDS or PDSE, due to the nature of these data structures.

## Creating a PDS or PDSE - `mkdir dsntype(pds)`, `dsntype(library)`

To create a PDS or PDSE, perform the following steps:

1. Create a local directory on your client to be used as a mount point. For example (with UNIX), enter this command:

```
$ mkdir /u/smith/mnt
```

2. Mount the MVS file system (accessing files that begin with the high-level qualifier of smith):

```
# mount mvshost1:"smith,mgmtclas(normal)"  
/u/smith/mnt
```

3. If creating a PDSE, use the **mkdir** (make directory) UNIX command, specifying the ***dsntype(library)*** attribute to create a PDSE named smith.data1ib:

```
$ mkdir /u/smith/mnt/"data1ib,dsntype(library)"
```

If creating a PDS, use the **mkdir** (make directory) UNIX command, specifying the ***dsntype(pds)*** attribute as follows:

```
$ mkdir /u/smith/mnt/"data1ib,dsntype(pds),dir(20)"
```

**Omitting *dsntype(pds)*:** You can omit specifying the ***dsntype(pds)*** attribute if pds has been specified for the ***dsntype*** attribute either in your site attribute data set or in your mount point.

4. You can use the **vi** UNIX command to create a PDS or PDSE member named smith.data1ib(member1):

```
$ vi "/u/smith/mnt/data1ib/member1,text"
```

Type your text, save it, and quit.

You have now created a PDS or PDSE member, which is processed in text processing mode. You can use the **cat** UNIX command to view the contents of your PDS or PDSE member.

**Note:** z/OS NFS server supports a maximum of 14,562 members in a PDS or PDSE data set. When a NFS read-directory request on a PDS or PDSE is processed, the z/OS NFS server will return up to 14,562 member names. Other requests, such as read and write, to individual members are not affected.

## Removing a PDS or PDSE - `rm`, `rmdir`

To remove a PDS or PDSE, first make sure that the PDS or PDSE is empty. You can delete all members under the directory using the **rm** UNIX command. Then use the **rmdir** (remove directory) UNIX command. This example removes the data1ib directory, and confirms its removal by a failed try to query it (**ls** is the UNIX list files command):



```
$ ls -F /u/smith/mnt/datalib
data1* data2* data3*
$ rm /u/smith/mnt/datalib/*
$ rmdir /u/smith/mnt/datalib
$ ls -F /u/smith/mnt/datalib
/u/smith/mnt/datalib not found
```

## Accessing PDS or PDSE members

There is more than one way to mount and access PDS and PDSE members. For example, you can display the existing PDS member smith.source(bigblue) by entering either of these command sequences:

```
$ mkdir /mnt
# mount hostname:"smith.source,text" /mnt
$ cat /mnt/bigblue
```

or

```
$ mkdir /mnt
# mount hostname:"smith,text" /mnt
$ cat /mnt/source/bigblue
```

These two approaches are equivalent.

## Updating or extending a PDS or PDSE member

The z/OS NFS server does not support updating or extending a PDS or PDSE member directly. To update or extend a PDS or PDSE member, a client program must follow these steps:

1. Copy the file to the client machine.
2. Update or extend the copied version on the local system.
3. Truncate the original MVS file to zero size by sending a SETATTR request with zero file size.
4. Copy the updated version on the local host to MVS by writing request.

Some client editors follow the above steps, for example, OS/2 EPM and KEDIT editors, and the AIX and UNIX vi editor. Other editors do not follow the above steps, for example, OS/2 E editor and the z/OS UNIX OEDIT editor. In the latter case the user must save the updated version into a new file.

## Timing out while writing a PDS or PDSE member

If you are writing to a PDS or PDSE member and a timeout occurs, the timeout causes the member to close. The remaining write requests appear to append to a PDS or PDSE member. The write request does not complete successfully and causes an I/O error. To avoid timing out, increase the time on the timeout setting.

## Wildcard copy to a PDS or PDSE

To ensure that a wildcard copy of a PDS or PDSE is completed successfully, the PDS or PDSE member must be closed and dequeued (if necessary). For example, using the statement below, a wildcard copy will fail if any member inside of smith.datalib is open:

```
$ cp smith.* /u/smith/mnt/datalib
```

## Limitations of a PDS

The PDS support in the NFS adheres to the conventions used in MVS. For example, you cannot have more than one member of a PDS open for output at a time. If you try to CREATE | REMOVE | RENAME | WRITE a member of a PDS while another member is open for output, you get a “Permission denied” message.

A PDS member stays open for the timeout period specified in the appropriate timeout processing attribute, or until you try to create or write to another member.

## Concurrent writes to a PDSE

The NFS supports concurrent writes to a PDSE. If you are writing to one member of a PDSE, another client can write to any other member in the same PDSE. You can also use ISPF to edit a PDSE member while another NFS client is writing to a different member of the same PDSE.

---

## Creating VSAM files

The NFS supports three types of VSAM files: key-sequenced (KSDS), entry-sequenced (ESDS), and relative record (RRDS). However, keyed access and relative-number access to the files are not supported.

If you plan to update a VSAM data set (for example, with the **vi** editor or with the **cp** copy command), the data set must have been defined with the **reuse** option. Trying to write back a VSAM data set that was not defined as reusable results in an “I/O error”, “failure to open”, or similar error message. If you create a VSAM file using the NFS, the **reuse** option is used by the server.

For more information on VSAM files, see *z/OS DFSMS: Using Data Sets*.

In the following example, the attributes indicate that:

- Spanned records are allowed
- Organization is key-sequenced
- Keys are 8 bytes long and start in position 0 of each record
- Average record size is 1024
- Maximum record size is 4096
- Space is allocated for 50 records with a secondary allocation of 10
- Cross-region and cross-system share options are provided
- The file is to be created on a volume named D80CAT

```
$ cp ksds.old "ksds.new2,spanned,dsorg(indexed),keys(8,0),
recordsize(1K,4K),space(50,10),shareoptions(1,3),
vol(D80CAT)"
```

In the following example for creating a VSAM ESDS file, the attributes indicate that:

- Spanned records are allowed
- Organization is entry-sequenced
- Average record size is 1024
- Maximum record size is 4096
- Space is allocated for 50 records with a secondary allocation of 10
- Cross-region and cross-system share options are provided
- The file is to be created on a volume named D80CAT

```
$ cp esds.old "esds.new3,spanned,dsorg(nonindexed),
  recordsize(1K,4K),space(50,10),shareoptions(1,3),
  vol(D80CAT)"
```

In the following example, the attributes indicate that:

- Spanned records are not allowed
- Organization is relative record, numbered in ascending order
- Average record size is 1024
- Maximum record size is 1024
- Space is allocated for 50 records with a secondary allocation of 10
- Cross-region and cross-system share options are provided
- The file is to be created on a volume named D80CAT

```
$ cp rrds.old "rrds.new4,nonspanned,dsorg(numbered),
  recordsize(1K,1K),space(50,10),shareoptions(1,3),
  vol(D80CAT)"
```

---

## Exploiting SAM striped files

With SAM striping, data I/O is done in parallel to improve performance. For a file with 16 stripes, data is simultaneously processed on the first track of the allocated space in each of the 16 volumes. This allows for quick access to all the information.

The z/OS NFS server can support data set striping through the use of data class and storage class attributes that define extended format data sets. The z/OS NFS server can exploit the performance of extended format data sets by reading multiple blocks at a time when reading ahead.

For more information about striped files, see *z/OS DFSMS: Using Data Sets*.



---

## Chapter 4. Using z/OS UNIX System Services files

This chapter explains what you need to know to access z/OS UNIX files from a client workstation. This chapter explains:

- The hierarchical file system (HFS)
- POSIX compatibility
- Attributes specific to z/OS UNIX
- Protecting your z/OS UNIX files
- Accessing z/OS UNIX files from the client
- Linking an MVS data set to a hierarchical file system
- UNIX look and feel

**Note:** For detailed information about z/OS UNIX, see *z/OS UNIX System Services User's Guide*.

---

### HFS file system

z/OS UNIX provides a hierarchical file system (HFS) for MVS. A file within z/OS UNIX is called an HFS file. HFS files are organized in a hierarchy of files and directories on a tree much like UNIX. A directory can contain files or other sub-directories. The highest level directory is called the root directory. Figure 3 on page 30 shows an example of mounting an HFS directory from a UNIX client.

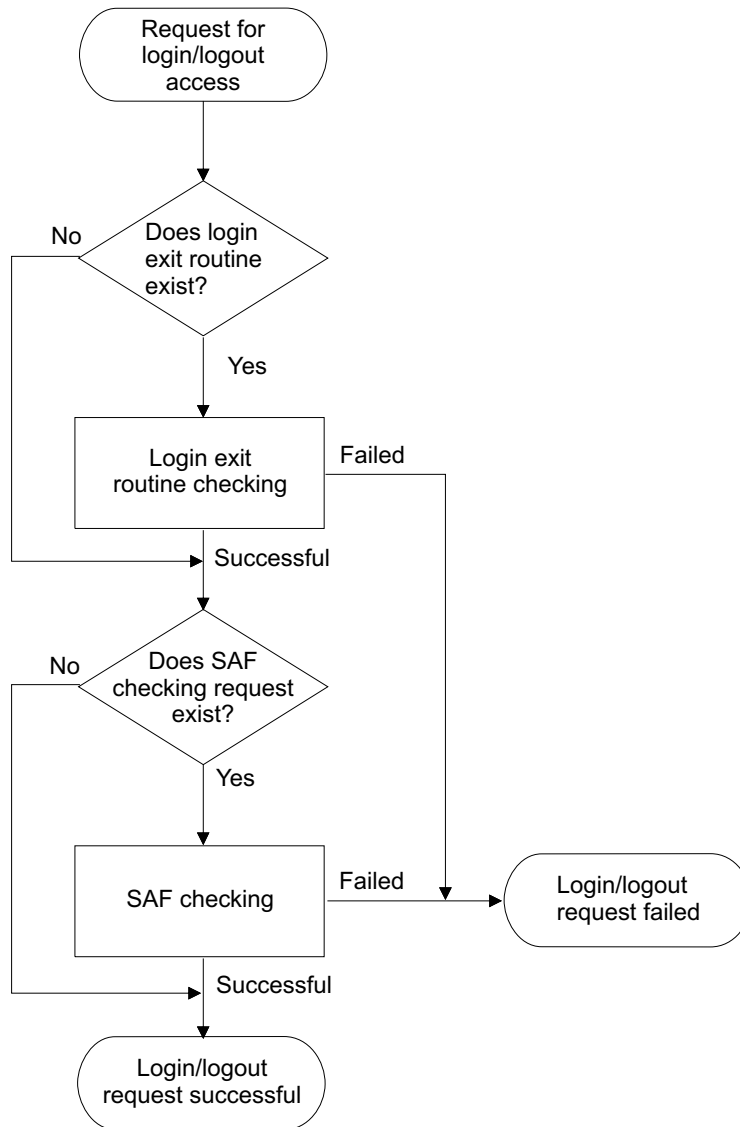


Figure 3. Example of mounting an HFS file from a UNIX client

An HFS file system must be mounted by an MVS system operator using a TSO **mount** command before that HFS file system can be mounted by an NFS client via the z/OS NFS server.

HFS files are byte-oriented rather than record-oriented (unlike conventional MVS data sets). This data can be shared with TSO z/OS UNIX users in addition to NFS clients. All data written to an HFS file can be read by all programs as soon as it is written. You can also copy data between HFS and MVS data sets using z/OS UNIX utilities like ISHELL.

## Text or binary processing

HFS is a byte-oriented, hierarchical, EBCDIC file system. The z/OS NFS server provides ASCII to EBCDIC text translation. If you are just using the mainframe as a repository for your workstation (ASCII) data, you should use the binary mode to speed processing. If you use text mode, data from your workstation is converted into EBCDIC when it is stored on the mainframe. Conversely, when the z/OS NFS server returns the data to your client system, it converts the data back into ASCII.

The conversion can slow processing, but might be necessary if you are sharing data with other MVS users. All data is converted according to the active translation table. Therefore, if the data set contains a mixture of characters and binary data, binary data is converted as well. In text mode, then, be careful not to mix your text data (characters) with binary data.

In text mode, you can either use the OEMVS311 translation table or a customized translation table to convert data between ASCII and EBCDIC. If you are using z/OS UNIX and text mode processing, specify the OEMVS311 translation table with the `xlat` processing attribute. The OEMVS311 table translates ASCII (ISO 8859-1) to and from EBCDIC (1047 - z/OS UNIX System Services). TCP/IP for MVS version 3.1 provides the OEMVS311 table. This table translates the UNIX line terminator (`\f`) to the z/OS UNIX line terminator (`\n`). See *TCP/IP for MVS: Customization and Administration Guide* for more information about creating and customizing your own translation tables.

This is an example of specifying the OEMVS311 translation table during a mount:

```
$ mount 1stc3mvs: "/hfs/usr/man/C,text,xlat(oemvs311)" /mnt
$ export MANPATH=/mnt
$ man more
```

In this example:

- `1stc3mvs` is the name of the z/OS UNIX host
- `/hfs` is the HFS prefix
- `/urs/man/C` is the HFS directory to be mounted
- `text` specifies that data be converted between ASCII and EBCDIC
- `xlat(oemvs311)` specifies that the translation table named OEMVS311 is used to convert data between ASCII and EBCDIC
- `/mnt` is the local mount point
- `man more` obtains a Man Page description of the **more** command

---

## POSIX compatibility

The NFS supports file access to the z/OS UNIX file system. z/OS UNIX supports a set of standards called the portable operating system interface (POSIX). See *z/OS UNIX System Services User's Guide* for more information about POSIX compliance. With z/OS UNIX, the NFS performs the following functions:

- Supports hierarchical directories
- Allows file names up to 255 characters in length
- Allows path names up to 1023 characters in length
- Supports mixed-case names and special characters, except nulls, slashes, and commas in file and path names
- Supports UNIX-style file access permissions
- Supports group ID and user ID at the file level
- Supports the full NFS protocol (including external links)
- Enables data sharing between clients and the z/OS UNIX
- Enables you to link conventional MVS data sets to a POSIX path name

This support incorporates the basic strengths of the MVS/ESA system for both existing MVS data and applications and for new POSIX conforming data and applications.

---

## NFS protocol

The z/OS UNIX are compliant with all of the z/OS Network File Systems version 2 and version 3 protocols.

---

### Attributes specific to z/OS UNIX System Services

The following attributes are specific to the z/OS UNIX:

sync	Processing attributes for version 2 protocol only
async	Processing attributes for version 2 protocol only
hfs	Site attribute
extlink	Attribute, see “Linking an MVS data set to a hierarchical file system” on page 34

**Note:** These attributes are also explained in “Chapter 10. Attributes for the z/OS NFS server” on page 75.

### Synchronous write to an HFS file for NFS version 2 protocol

Use the **sync** and **async** processing attributes to specify whether data received by a write request for an HFS file object is committed to nonvolatile storage before the write response is returned to you.

If **sync** is specified for an HFS file object, the data is written to HFS.

For greater throughput, you can alternatively specify **async**. Your data is then committed to the disk some time after the write request is received from the NFS client. Your data is written to disk when the write timeout occurs, or if z/OS UNIX reclaims buffer cache storage.

The **sync** and **async** processing attributes only apply to HFS data access. They are ignored for MVS data set access. A TSO z/OS UNIX user doesn't have to wait for the data to be committed to nonvolatile storage before accessing. z/OS UNIX maintains a central buffer cache and a TSO z/OS UNIX user can access the data as if it were in the file once it has been written by the z/OS NFS server.

### Synchronous write to an HFS file for NFS version 3 protocol

For the write procedure there is a processing argument **stable** and output parameter **commit** which specifies whether data received by a write request for an HFS file object is committed to nonvolatile storage before the write response is returned to you.

If the **stable** processing argument is used during the write procedure, there are three modes when the write procedure writes to a file:

**file\_sync** - The z/OS NFS server must commit all data written and all file system data to stable storage before returning commit results.

**data\_sync** - The z/OS NFS server must commit all data written and sufficient metadata to enable retrieval of data, before it returns a reply to the client.



**unstable** - The z/OS NFS server may not commit any part of the data and metadata to stable storage, before returning a reply to the client. The data will be committed when a timeout occurs.

For the commit procedure, the z/OS server will support committing the entire data and metadata to stable storage.

## Authorization checking when writing to an HFS file

There is no support of OPEN and CLOSE of a file since SUN NFS Protocol is a stateless model. In order to support the OPEN and CLOSE semantics, the z/OS NFS server allows the owner of an HFS file to write permission regardless of the UNIX permission bits setting on the file.

## HFS site attribute

To access z/OS UNIX files, you must know the HFS prefix defined by your system administrator (the default is /hfs). You can use the **showattr** command to display the HFS prefix defined for your location. You use this prefix in your mount command before the path name of the z/OS UNIX directories that you are mounting. The HFS prefix is used by the NFS server to distinguish z/OS UNIX directories from conventional MVS data sets, but the HFS prefix isn't part of the path name that you see. After you have entered the mount command, you access HFS files using the local mount point.

---

## Protecting your z/OS UNIX System Services files

As an z/OS UNIX user, you can control the read, write, and execute access to your files by other users in and outside of your group by setting the permission bits associated with the files.

To access z/OS UNIX files from the NFS, you must be defined as an z/OS UNIX user. The system programmer defines you as an z/OS UNIX user by assigning an z/OS UNIX user ID (UID) and an z/OS UNIX group ID (GID) to you. The UID and GID are numeric values associated with a TSO/E user ID. The values are set in the RACF user profile and group profile when you are authorized to use z/OS UNIX. The system uses the UID and GID to identify the files that you can access. Your specific UID value identifies you as a user of z/OS UNIX services. A GID value is a unique number assigned to a group of related users. These numbers appear in the RACF user profile. See *z/OS UNIX System Services Planning* for more information.

---

## Accessing z/OS UNIX System Services files from a client

Most of the commands that are used to access z/OS UNIX files are identical to the commands that are used to access conventional MVS data sets. These are the commands:

mount	mvslogout	umount
mvslogin	showattr	

The only command that is changed for z/OS UNIX is the **mount** command.

**Note:** The syntax of the commands listed above may vary between platforms, see the appropriate chapter for examples specific to the platform you are using to access z/OS UNIX files.

If you are using AIX (or any other UNIX-based operating system) see “Chapter 6. Commands and examples for AIX and UNIX clients” on page 39.

If you are using DOS see “Chapter 7. Commands and examples for Sun PC-NFS clients” on page 49.

## Mount examples

Table 9 shows how to mount z/OS UNIX files from various platforms.

Table 9. Examples of the mount command for clients

Clients	Command Examples
AIX, UNIX, Solaris	mount mvshost1:”/hfs/smith” /u/smith/mnt
DOS	mount z: mvshost1:/hfs/smith
Sun PC-NFS	net use z: mvshost1:/hfs/smith

In Table 9:

- mvshost1 is the name of the MVS host
- /hfs is the HFS prefix
- /smith is the HFS directory to be mounted
- /u/smith/mnt is the local mount point

**Note:** The /hfs prefix value is used by the z/OS NFS server to determine if a file is an z/OS UNIX file, and does not appear in the path name of an HFS file once it is mounted.

---

## Linking an MVS data set to a hierarchical file system

This section explains how to access an MVS data set via an HFS path name by using the external link command. It also explains how to display the contents of an external link and how to delete an external link.

### Creating an external link

You can create an external link to an MVS data set, and then transparently access the MVS data set by referencing the external link. The external link simulates a UNIX-like hierarchical naming convention for conventional MVS data sets. This is done using the **ln** command, for example:

```
mount mvshost1:USER1 /mnt
mount mvshost1:/hfs/u/nfs /samples
ln -s USER1.MVSFILE /samples/linkfile,"extlink"
```

In this example an HFS file object, /linkfile, of the file type “extlink” is created containing the file name of the MVS data set USER1.MVSFILE to be accessed. The source file must be mounted to an HFS file system. The external link must reference an MVS data set. All future references to /samples/linkfile access USER1.MVSFILE transparently.

In this example the file /usr/pub/myfile is copied to the MVS data set USER1.MVSFILE that is contained in the external link /samples/linkfile:

```
cp /usr/pub/myfile /samples/linkfile
```

Your installation should make sure that the appropriate security permissions have been obtained to access the MVS data set. You will receive "Permission Denied" message if the mount point /mnt has not been established on USER1.

A mount point must be established before the external link is established. Otherwise, the error code ACCESS DENIED is returned. For physical sequential data sets, the high level qualifier of a data set must be established. For example, if you had a file called smith.test.data you can mount with smith, smith.test, or smith.test.data as your high level qualifier. For PDS and PDSE data sets, the fully qualified name must be established as a mount point. An example of a fully qualified name would be, smith.test.data.

## Displaying the contents on an external link

You can display the contents of an external link by appending the "extlink" sequence to the external link path name. This permits the user to inspect the contents of the external link with the **ls -l** command.

This example shows how to display the attributes and contents of the external link /samples/linkfile:

```
ls -l /samples/linkfile,"extlink"  
lrwxrwxrwx 1 user1 13 Jun 17 20:43 /samples/linkfile ->USER1.MVSFILE
```

This example shows how to display the attributes of the MVS target data set USER1.MVSFILE:

```
ls -l /samples/linkfile  
-rw-rw-rw- 1 root 2112 Sep 28 13:50 /samples/linkfile
```

## Deleting an external link

The external link file object is deleted with the remove request:

```
rm /samples/linkfile,"extlink"  
rm /samples/linkfile
```

Either **rm** command results in the HFS external link file object alone being removed. The target MVS data set is not affected.

---

## UNIX look and feel

Using the z/OS NFS with z/OS UNIX managed files provides UNIX client users with a transparent view of their data. The file attributes are maintained in the same way as is found on any UNIX system:

- Regular, directory, link, device, and FIFO file types
- User, group, and other read/write/execute access permissions
- UID and GID file ownership
- File size

To access HFS files, it is necessary to be defined to RACF as an z/OS UNIX user. Some installations might prefer to provide users with unrestricted access to their z/OS UNIX data by specifying security(none) or security(exports) in the site attributes. With this setting, the client's user ID and group ID credentials are used for all file access authentication, and there is no requirement for the user to be defined to RACF or to perform the mvlogin command.

**Note:** For security(none) and security(exports) options, the UID of the root (UID=0 from the workstation) is mapped to UID of NOBODY (UID=65534) by the z/OS NFS server. The implication is that the z/OS NFS server will use the mapped UID of 65534 for all HFS file authorization checking. For example, file creation owner UID is set to 65534 in the HFS file attribute.

## File system size

For HFS files, see Table 10 and Table 11 for the following file system values that are returned:

*Table 10. File system values to get static file system information (fsinfo)*

Value	Description	HFS Value
rtmax	Maximum number in bytes for the read request supported by the server	65536 (64KB)
rtpref	Preferred size of the read request	32768 (32KB)
rtmult	Suggested multiple for the size read request	4096
wtmax	Maximum size of a write request supported by the server	65536 (64KB)
wtpref	Preferred size of the write request	32768 (32K)
wtmult	Suggested multiple for the size of a write request	4096
dtpref	Preferred size of the readdir request	8192

*Table 11. File system values to retrieve POSIX information (pathconf)*

Value	Description	HFS Value
linkmax	Maximum number of hard links	2 ** 31
name_max	Maximum length of a component file name (file name + attributes)	255
no_trunc	Server will reject any name that is longer than the name_max	True
chown_restricted	Change either the owner or the group associated with the data set	False
case_insensitive	Server does not distinguish the case when interpreting file names	False
case_preserving	If True, the server file system will preserve the case of a name during a create, mkdir, mknod, symlink, rename, or link	True

---

## Chapter 5. Network Lock Manager (NLM) and Network Status Monitor (NSM)

This chapter provides an overview of the z/OS NFS Network Lock Manager (z/OS NFS NLM) and the z/OS NFS Network Status Monitor (z/OS NFS NSM). It explains how they work together to provide file locking and access control capability over z/OS NFS. In addition, this chapter also explains the following features:

- Monitored lock
- Non-monitored locks
- Locking files
- Locking records

---

### Using Network Lock Manager (NLM)

The z/OS NFS NLM allows a client on the host to lock a record or a file on the z/OS NFS server. A client user can either choose to lock the entire file or a record section of a file. The two types of locks that the client host uses are monitored locks and non-monitored locks.

The z/OS NFS NLM only supports advisory locking. Advisory locking is when the operating system keeps track of which files have been locked by which process, but does not prevent a process from writing to a file that is locked by another process. This means that a process can ignore an advisory lock if the process has adequate permission.

#### Monitored locks

Monitored locks provide the client user with reliability. If the server host on which the monitored locks are established fails, the locks are reinstated when the server host recovers. The locks that are held by the client host are discarded by the z/OS NFS NLM on the server host if the client host fails before the locks are released. Monitored locks will only work correctly if both the server host and the client host are running NSM.

#### Non-monitored locks

Non-monitored locks are used on personal computer (PC) operating systems. Non-Monitored locks provide the same functionality as the monitored locks with one exception. If the server host on which the locks are established, fails and recovers, the locks will not be re-established. The client host is responsible for detecting a server host failure and re-establishing the locks. In addition, the client host informs the z/OS NFS NLM when it has rebooted so that the client host can discard all of the locks and file shares held for the client.

#### Locking records

A record is a byte-stream range. The record starts at the current offset and extends forward for a positive size value and backwards for a negative size value. If the size value is zero, the record affected extends forward from the current offset through the end of the file.

In order to lock a record, the client user can use the *lockf* function. The *lockf* function has four values. The F\_ULOCK value unlocks a previously locked region. The F\_LOCK value locks a region. The F\_TLOCK value tests and locks a region. The F\_TEST value tests a region to see if it is locked. When the client user uses

the `F_LOCK` function when the region is being accessed by another function, the `F_LOCK` function waits for the region to become available and then perform the locking function. This is called blocking. When the client user uses the `F_TLOCK` function and the region that it is being locked is being used by another function, `lockf` returns with a value of -1 and `errno` set to `EAGAIN` or `EACCESS`. This is called nonblocking. By using the `F_TEST` function, the client user can check to see if a lock is set, without setting a lock.

If the file starts with the size value of zero and the `lockf` system call is used, then the entire file is locked. In order to issue a byte range lock, the client user must specify a small program that specifies the section of the file that they are locking. In “Appendix E. Sample program for `lockd` and `statd`” on page 151, the client user is issuing a byte range lock.

## Locking a file

The `flock` system call is used to lock and unlock a file. The `flock` system call uses four constants to lock a file. The `LOCK_SH` constant specifies shared locks while the `LOCK_EX` specifies exclusive locks. The `LOCK_UN` constant unlocks the file and the `LOCK_NB` specifies that the file should not be blocked when it is locked. The `LOCK_NB` can be used with both the shared locks and the exclusive locks.

More than one shared lock can be applied to a file but the file cannot have both a shared lock and an exclusive lock at the same time. Furthermore, a file also cannot have multiple exclusive locks at one time. If the lock is successful, zero is returned by `flock`. If the lock is unsuccessful, then a -1 is returned. When the lock fails because the file is already locked and a nonblocking lock was requested, the `errno` is set to `EWOULDBLOCK`. See “Appendix E. Sample program for `lockd` and `statd`” on page 151.

---

## Using Network Status Monitor (NSM)

The z/OS NFS NSM is a service that provides applications with information on the status of network host. Each z/OS NFS NSM keeps track of its own “state” and notifies any interested parties of a change in its “state”.

For correct operation of the z/OS NFS NSM, the client and the server hosts are required to monitor each other. When a lock request is issued by a process running on the client host, the NLM on the client host request the NSM on the client host to monitor the server host. The client NLM then transmits the lock request to the z/OS NFS NLM on the server. On reception of the lock request the z/OS NFS NLM on the server host will request the z/OS NFS NSM on the server host to monitor the client host. In this way, each host is monitored by the NSM on the other host.

## Chapter 6. Commands and examples for AIX and UNIX clients

This chapter gives the syntax and examples of commands that AIX RS/6000 users need to know to access MVS data sets from a client. Some examples are also provided for HP/UX, UNIX, Sun Solaris and SunOS environments. This chapter shows how to perform the following tasks:

- Log on to MVS from your client.
- Access MVS files from your client.
- Display default mount point attributes.
- Query mount points.
- Unmount MVS files from the client.
- Log out of MVS.

The **mount** and **umount** commands are operating system specific commands. They are not shipped with z/OS NFS. See your NFS client documentation for the exact syntax and usage.

### Using commands on AIX

Figure 4 shows a summary of the syntax for the commands described in this chapter:

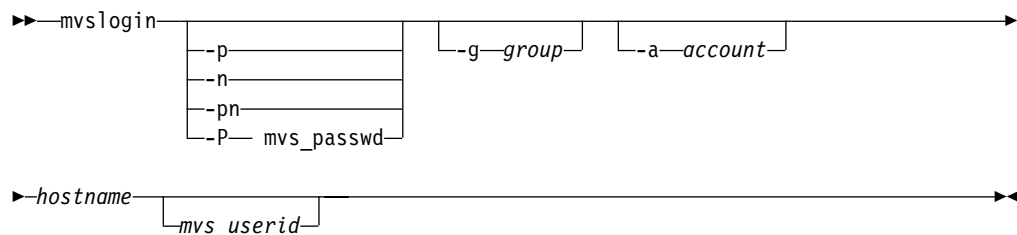


Figure 4. Syntax for commands

-p	Causes a prompt for your MVS password. The password is passed to MVS to validate the user logging in. Your security procedures determine whether you should use this parameter.
-n	Causes a prompt for a new password.
-pn	Causes a prompt for the user's current password and then causes two prompts for the user's new password.
-P mvs_passwd	No prompt for your MVS password; just type your MVS password after the -P. This enables you to automate your MVS login.
-g group	A group name string passed to MVS for accounting purposes. The maximum length is 8 characters.
-a account	An account string passed to MVS for accounting purposes. The maximum length is 16 characters.
hostname	The name of the MVS host (for example, mvshost1).
mvs_userid	A user ID that MVS recognizes as valid. If you do not specify this parameter, your workstation user name is used. The z/OS NFS server does not support the use of an alias user ID or a mixed case user ID with the <b>mvslogin</b> command.

```

▶▶—mount—[—o clnt_opt—]—hostname:"mvs_qual"—[—,attribute—]"—/localpath—▶▶

```

<i>-o clnt_opt</i>	The client <b>mount</b> command options (such as <code>soft,timeo=20</code> ). Refer to the documentation of your client operating system for a description of the options for your client environment.
<i>hostname</i>	The name of the MVS host (for example, <code>mvshost1</code> ).
<i>mvs_qual</i>	The path name of an z/OS UNIX file (must begin with the /HFS prefix) or the high-level qualifier(s) of a conventional MVS data set that you are accessing in the MVS file system (such as <code>smith</code> or <code>smith.project</code> ).
<i>attribute</i>	A z/OS NFS server data set creation or file processing attribute (such as <code>text</code> ). See "Chapter 10. Attributes for the z/OS NFS server" on page 75. If you specify any attributes, make sure you enclose <i>mvs_qual</i> and the attributes in double quotation marks.
<i>/localpath</i>	The mount point on your client system (for example, <code>/u/smith/mnt</code> ). This should be an empty directory.

```

▶▶—showattr—[—t—]—hostname—[—/localpath—]—▶▶

```

<i>-t</i>	Used to specify tersed output.
<i>hostname</i>	The name of the MVS host (for example, <code>mvshost1</code> ).
<i>/localpath</i>	The mount point on your client system (for example, <code>/u/smith/mnt</code> ). This should be an empty directory.

```

▶▶—umount—/localpath—▶▶

```

<i>/localpath</i>	The mount point on your client system (for example, <code>/u/smith/mnt</code> ). This should be an empty directory.
-------------------	---

```

▶▶—mvslogout—hostname—▶▶

```

<i>hostname</i>	The name of the MVS host (for example, <code>mvshost1</code> ).
-----------------	---

## Quick reference of AIX and UNIX commands

This screen is an example of a standard MVS login and logout procedure for AIX RS/6000:

- `smith` - MVS user ID and high level qualifier for MVS data sets
- `mvshost1` - System name of MVS host
- `/u/smith/mnt` - The name of the mount point
- `GFSAnmnt` - Messages starting with GFSAn apply towards z/OS NFS requests. These messages are explained in "Appendix A. Network File System messages" on page 93.



```
# mvslogin mvshost1 smith
Password required
GFS9731 Enter MVS password: password
GFS9551 smith logged in ok.
# mount mvshost1:"smith" /u/smith/mnt
mount: mvshost1:"smith"
"smith" was attached successfully.
# umount /u/smith/mnt
Unmounting '/u/smith/mnt' ... successful
# mvslogout mvshost1
UID 200 logged out ok.
```

You can use the **mount** command with no parameters specified to list the mount points on your client system.

---

## Accessing z/OS UNIX System Services and conventional MVS

To access z/OS UNIX files or conventional MVS data sets, enter both the **mvslogin** command to log into the MVS host system and the **mount** command to mount the files or data sets to your local system. The **mvslogin** command is only required when accessing data on systems where the z/OS NFS server site security attribute is set to **saf** or **safexp**. Once the files or data sets are mounted to a local directory, you can read, write, create, delete, and treat the mounted files as part of your workstation's local file system. When you are finished with your work, use the **umount** and **mvslogout** commands to break the connection. The **mvslogout** command is only required when the **mvslogin** command was used to begin the connection.

To access files on MVS systems where the z/OS NFS server site security attribute is set to **saf**, **exports**, or **safexp**, you need an MVS user ID and password, and authorization to access the files that you need. You can only establish one MVS session for each MVS user ID. If you do not already have an MVS user ID, an MVS password, and access authorization, request them from your MVS system administrator.

**Note:** If you cannot use the **mvslogin**, **mvslogout**, or **showattr** commands, they might be installed incorrectly or in another directory. Ensure that your system administrator has made the executable code for these three commands available to your workstation and that you have been given the correct path name to find the commands. Also, make sure that your version of these commands matches the release of the z/OS NFS that you are using. Otherwise, the commands might not function properly.

### Mvslogin command examples

Use the **mvslogin** command to log in to MVS from your workstation. The **mvslogin** command can be issued multiple times and the last one overrides the previous one. The **mvslogin** command is only required when accessing data on systems where the z/OS NFS server site **security** attribute is set to **saf** or **safexp**.

Table 12 shows examples of the **mvslogin** command where mvshost1 is the name of the MVS host and smith is the user's ID on MVS.

*Table 12. Examples of the mvslogin command for clients*

---

```
mvslogin -p mvshost1 smith
```

---

```
mvslogin -P smithspw -g finance -a 5278 mvshost1 smith
```

---

Table 12. Examples of the mvlogin command for clients (continued)

mvlogin -n mvshost1 smith
mvlogin -pn -a 5278 mvshost1 smith
mvlogin mvshost1
mvlogin mvshost1 smith

In the example where the user enters mvlogin mvshost1, the current login client user ID is used as the MVS user ID.

In the example where the user enters mvlogin mvshost1 smith, the system then prompts for Smith's MVS password. If Smith logs in successfully, this message appears:

GFSA955I smith logged in ok.

Otherwise, an appropriate error message appears.

**Note:** Messages with the prefix of GFSA and GFSC apply to NFS requests. These messages are further explained in "Appendix A. Network File System messages" on page 93.

When an z/OS UNIX UID or GID segment is defined with the user identification, an additional check is done to compare the z/OS UNIX UID or GID with the client UID or GID during the login processing. An informational message is returned when the server and the client UID or GID do not match. This informational message contains the z/OS UNIX UID and GID for the MVS user identification.

**Note:** The authentication is considered successful even if the UID and GID do not match. The message is issued for the user's information only.

For the PCNFSD authentication request, the server UID and GID is returned to the client user if the UID and GID are defined. Otherwise, an arbitrary number is generated and returned to the client user.

### "Permission denied" message

If you have successfully logged in and get the "Permission denied" message while trying to access data, that can be due to one of the following cases:

- An **mvlogout** command for the same MVS host has been entered from the same client platform. See mvlogout for details.
- Your MVS user ID has been automatically logged out because the **logout** attribute value has been exceeded. This can happen when you leave the client idle for too long. Enter the **mvlogin** command again, and start your processes again. To find out how many seconds you can stay logged in while your client is idle, issue the **showattr** command and look at the **logout** attribute.
- Another mvlogin to the same MVS host using the same MVS ID has been entered from the same UID and the same client platform. If this is the case, retry your access.

**Note:** Some clients give a somewhat different message such as "Access is Denied".

## Mount command examples

Use the **mount** command to make a connection between a mount point on your local file system and one or more files in the MVS file system.

```
# mount mvshost1:"smith" /u/smith/mnt
```

In this example:

- mvshost1 - The name of the host server
- smith - The name of the high-level qualifier of the MVS files
- /u/smith/mnt - The name of the mount point (preferably an empty directory)

At the same time, you can also predefine some of the attributes for the mount point, overriding the default attributes.

### Overriding default attributes

To override the default attributes, specify different attributes with the **mount** command or in a file access or creation command (such as **vi** in AIX or UNIX).

There are two kinds of attributes that you can modify:

**Data set creation** - Attributes provide information about an MVS file to the z/OS NFS server, such as:

- The type of file
- How the file is allocated

**Note:** Data set creation attributes do not apply to z/OS UNIX files.

**Processing** - Attributes provide information to the z/OS NFS server about how to handle the file. For example:

- How long the files remain open
- Whether the file contents are sent and received in text form, or in binary form to avoid ASCII/EBCDIC conversion

Use the **showattr** command to display the default data set creation and processing attributes. For descriptions of the attributes, see "Chapter 10. Attributes for the z/OS NFS server" on page 75.

Files are created and processed using the mount point attributes that were in effect when the files were last mounted. If your installation's default attributes have been changed (via the **exportfs** operand of the **modify** system operator command or restart of the server) and you want to apply these new default attributes, you can unmount and mount again (using the **umount** and **mount** commands).

When you access the file with a data access or creation command, you can override some of the attributes that were set by a **mount** command or the server default settings.

## Using mount to override server default attributes

In this example, the **mount** command is used to modify two processing attributes, specifying **binary** (rather than **text**) and **readtimeout(30)** (rather than the server default **readtimeout** value):

```
# mount mvshost1:"smith,binary,readtimeout(30)" /u/smith/mnt
```

In this example:

- mvshost1 - The name of the host server
- smith - The name of the high-level qualifier of the MVS files
- binary - The processing mode for file contents sent between MVS and the client (binary mode avoids ASCII/EBCDIC conversion)
- readtimeout(30) - The amount of time (30 seconds) allowed since the last read access before the file is closed
- /u/smith/mnt - The name of the mount point (preferably an empty directory).

### Getting authorization to access files

If the mount fails, check with your system administrator to ensure that you are authorized to access the MVS data sets or z/OS UNIX files and that the data sets or files are listed in the exports data set. The privilege level required to enter **mount** and **umount** commands varies among client operating system implementations. Many UNIX implementations limit these commands to the root user or superuser mode. If the MVS system operator issues the **freeze=on** operand of the **modify** command, all new tries to mount an MVS or HFS data set fail until the MVS system operator issues the **freeze=off** operand. If the z/OS UNIX system operator issues the **freeze=onhfs** operand of the **modify** command, conventional MVS data sets can still be mounted, but all new tries to mount z/OS UNIX files fail until the system operator issues the **freeze=offhfs** operand.

### Saving of mount points

Once the **mount** command is issued successfully and a mount point is established between a local directory and the MVS or z/OS UNIX file system, the mount point information is saved in the mount handle data set by the z/OS NFS server. This information is used to automatically reestablish active mount points when the server is started. When accessing conventional MVS data sets, a mount point is active if the last activity against this mount point is less than the **restimeout** attribute value set by the system administrator (restimeout does not apply for HFS file system access). The **mount** command does not need to be reissued for the same mount point in further sessions unless the **umount** command has been used to disconnect the mount point or the mount point has been disconnected by the cleanup activity of the **restimeout** site attribute. For more information about the **restimeout** site attribute see "Chapter 10. Attributes for the z/OS NFS server" on page 75.

### Automatic timed logout - logout attribute

If there is no activity on the client within the period specified in the **logout** attribute of the attributes file, or the server stops, the connection between the server and the client workstation is logged out automatically. You must issue the **mvslogin** command again to get access to the MVS files.

---

## Displaying default and mount point attributes - showattr

Use the **showattr** command to display the default attributes or the attributes that have been set for a specific mount point. If you specify a mount point, **showattr** shows the attributes for the mount point, including the overriding values. For descriptions of the attributes, see "Chapter 10. Attributes for the z/OS NFS server" on page 75.

If you omit the hostname, you must specify the /localpath.

Table 13 shows examples of the **showattr** command for some clients.

*Table 13. Examples of the showattr command for clients*

Client Environments	Command Examples
AIX, UNIX	<code>showattr mvshost1 /u/smith/mnt</code>
z/OS	<code>showattr mvshost1 /u/smith/mnt</code>
SunOS	<code>showattr mvshost1 /u/smith/mnt</code>
Solaris	<code>showattr mvshost1 /u/smith/mnt</code>

Make sure that your version of the **showattr** command matches the release of the z/OS NFS that you are using. Otherwise, the z/OS NFS server attributes do not display.

These examples show different ways you can use the **showattr** and **mount** commands.

Figure 5 shows a **showattr** command with just the host name (mvshost1 in this example) specified. The default attributes for the server are displayed.

```

$ showattr mvshost1

z/OS NFS Server Data Set Creation Attributes:

lrecl(80)          recfm(fb)          blksize(3200)
space(10,10)       blks              dsorg(ps)
dir(5)             unit()            volume(nfs003)
recordsize(512,4K) keys(64,0)        nonspanned
shareoptions(1,3)  dsntype(pds)          norlse
mgmtclas()         storclas()

z/OS NFS Server Processing Attributes:

text              lf                blankstrip
nofastfilesize    retrieve           maplower
mapleaddot        executebitoff      setownerroot
attrtimeout(120)  readtimeout(90)   writetimeout(30)
sync              nofileextmap       xlat(OEMVS)
srv_ccsid(0)      cln_ccsid(0)       nordrverf
sidefile(f066440.nfs.mapping)

z/OS NFS Server Site Attributes (not modifiable):

mintimeout(1)     nomaxtimeout       logout(1800)
nfstasks(8,8)     restimeout(48,0)   hfs(/hfs)
bufhigh(32M)      readaheadmax(16K)  cachewindow(128)
percentsteal(20)  maxrdfsleft(32)    logicalcache(16M)
smf(none)         nopcnfsd           security(exports,exports,exports)
leadswitch        sfmax(2)           nochecklist
fn_delimiter(,)   readdirtimeout(30)

public(IBMUSER.WEBNFS,/HFS/)

```

*Figure 5. Displaying default attributes*

If you use the terse (-t) option, the attributes display like this:

```

$ showattr -t mvshost1
lrecl(80),recfm(vb),blksize(3200),space(10,10),blks,dsorg(ps),dir(5),unit(),
volume(nfs003),recordsize(512,4K),keys(64,0),nonspanned,shareoptions(1,3),
mgmtclas(),dsntype(pds),norlse,dataclas(),storclas()
text,lf,blankstrip,nofastfilesize,retrieve,maplower,mapleaddot,executebitoff,
setownerroot,attrtimeout(120),readtimeout(90),writetimeout(30),sync,nofileextmap
xlat(OEMVS),srv_ccsid(0),cln_ccsid(0),nordrverf,sidefile(f066440.nfs.mapping)
mintimeout(1),nomaxtimeout,logout(1800),nfstasks(8,8),restimeout(48,0),
hfs(/hfs),bufhigh(32M),readaheadmax(16K),cachewindow(128),percentsteal(20),
maxrdfsleft(32),logicalcache(16M),smf(none),nopcnfsd,
security(exports,exports,exports),leadswitch,sfmax(2),nochecklist,
fn_delimiter(,),readdirtimeout(30),public(IBMUSER.WEBNFS,/HFS/)

```

Figure 6 illustrates the **showattr** command being used to display the attributes for the MVS host named mvshost1 as well as the mount point, /u/smith/mnt.

Figure 6 illustrates the specified options. In the second **showattr** command, the client user specifies /u/smith/mnt in addition to mvshost1. This shows the user's specified settings at that mount point, rather than the default settings in the attributes data set.

```

# mount mvshost1:"smith,text,space(5,0),trks" /u/smith/mnt
$ showattr mvshost1

z/OS NFS Server Data Set Creation Attributes:

lrecl(8196)          recfm(vb)          blksize(0)
space(100,10)       blks              dsorg(ps)
dir(27)             unit()             volume()
recordsize(512,4K)  keys(64,0)          nonspanned
shareoptions(1,3)
mgmtclas()          dsntype(pds)         norlse
dataclas()          storclas()

z/OS NFS Server Processing Attributes:

binary              lf              blankstrip
nofastfilesize      retrieve        maplower
mapleaddot          executebitoff   setownerroot
attrtimeout(120)    readtimeout(90) writetimeout(30)
sync               nofileextmap    xlat(OEMVS)
srv_ccsid(0)        cln_ccsid(0)    nordrverf
sidefile(hlq.nfs.mapping)

z/OS NFS Server Site Attributes (not modifiable):

mintimeout(1)       nomaxtimeout    logout(1800)
nfstasks(8,8)       restimeout(48,0) hfs(/hfs)
bufhigh(32M)        readaheadmax(16K) cachewindow(128)
percentsteal(20)    maxrdfsleft(32)  logicalcache(16M)
smf(userfile)       nopcnfsd        security(exports,exports,exports)
leadswitch          sfmax(0)         checklist
fn_delimiter(,)     readdirtimeout(30)
public(NFSUSR3,/HFS/usr/lpp/internet/server_root/pub)

```

*Figure 6. Displaying default and mount point attributes (Part 1 of 2).* The client user changed the space(100,10), blks, and binary attributes to space(5,0), trks, and text for the mount point /u/smith/mnt, and then specified that mount point in the second **showattr** command.

```
$ showattr mvshost1 /u/smith/mnt

z/OS NFS Server Data Set Creation Attributes:

lrecl(8196)          recfm(vb)          blksize(0)
space(5,0)           trks              dsorg(ps)
dir(27)              unit()             volume()
recordsize(512,4K)  keys(64,0)         nonspanned
shareoptions(1,3)
mgmtclas()           dsntype(pds)        norlse
dataclas()           storclas()

z/OS NFS Server Processing Attributes:

text                 lf                 blankstrip
nofastfilesize       retrieve           maplower
mapleaddot           executebitoff      setownerroot
attrtimeout(120)     readtimeout(90)   writetimeout(30)
sync                 nofileextmap       xlat(OEMVS)
srv_ccsid(0)         cln_ccsid(0)       nordrverf
sidefile(hlq.nfs.mapping)

z/OS NFS Server Site Attributes (not modifiable):

mintimeout(1)        nomaxtimeout       logout(1800)
nfstasks(8,8)        restimeout(48,0)   hfs(/hfs)
bufhigh(32M)         readaheadmax(16K) cachewindow(128)
percentsteal(20)     maxrdforszleft(32) logicalcache(16M)
smf(userfile)        nopcnfsd           security(exports,exports,exports)
leadswitch           sfmax(0)           checklist
fn_delimiter(,)      readdirtimeout(30)
public(NFSUSR3,/HFS/usr/lpp/internet/server_root/pub)
```

Figure 6. Displaying default and mount point attributes (Part 2 of 2). The client user changed the space(100,10), blks, and binary attributes to space(5,0), trks, and text for the mount point /u/smith/mnt, and then specified that mount point in the second **showattr** command.

## Unmounting and logging out of MVS

This section describes the **umount** and **mvslogout** (or **mvslogut**) commands.

### Disconnecting your mount point - umount

Use the **umount** command to break the connection between the mount point on your client and the server. When you issue this client command, the file you were editing is released (written to DASD). You do not need to unmount after each session, unmount only when you no longer have a need to access the MVS file system. Check the documentation for your client operating system to ensure that you enter the **umount** command correctly.

Table 14 shows examples of the **umount** command for some clients:

- u/smith/mnt - The mount point on the local file system
- mvshost1 - The name of the MVS host system

Table 14. Examples of the umount command for clients

Client Environments	Umount Command Examples
AIX, UNIX	umount /u/smith/mnt
SunOS	umount mvshost1
Solaris	umount mvshost1

For example, suppose that you want to unmount from the server, and the mount point on your workstation is named /u/smith/mnt. You could enter the **umount** command as follows:

```
# umount /u/smith/mnt
```

“No Such File or Directory” Message - The MVS system operator can also unmount your workstation from the server. If this happens before you try to unmount, you get a “No such file or directory” error message.

Ending your MVS session - **mvlogout**, **mvslout**

Use the **mvlogout** or **mvslout** command to disconnect from the MVS host. The **mvlogout** command is only required when the **mvlogin** command was used to begin the connection.

An **mvlogout** to an MVS user ID cancels a prior **mvlogin** to the same MVS user ID from the same local host.

Your account is automatically logged out if it is inactive for the period of time specified in the **logout** site attribute.

Table 15 shows an example of the **mvlogout** command for some clients, in which the name of the MVS host is mvshost1.

Table 15. Example of the mvlogout command for clients

Client Environments	Mvlogout Command Examples
AIX, UNIX	mvlogout mvshost1
HP/UX	mvlogout mvshost1
SunOS	mvlogout mvshost1
Solaris	mvlogout mvshost1

If you log out successfully, a message like this appears:

```
GFSA958I uid 215 logged out ok.
```



## Chapter 7. Commands and examples for Sun PC-NFS clients

This chapter gives the syntax and examples of commands that you need to know to access files residing on MVS from your DOS client. This chapter shows how to perform the following tasks:

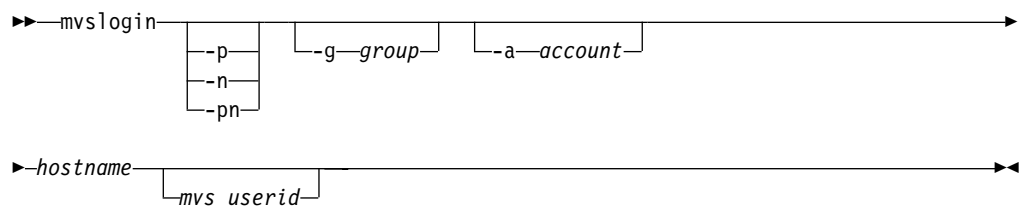
- Log on to MVS from your client.
- Access MVS files from your client.
- Display default mount point attributes.
- Unmount MVS files from the client.
- Log out of MVS.

In a DOS environment, you can use the SUN PC-NFS system.

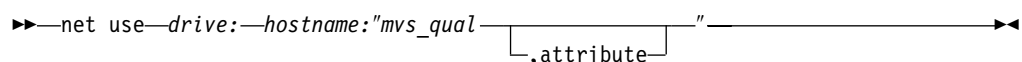
The **mount**, **umount**, and **net use** commands are operating system specific commands. They are not shipped with the NFS. See your Network File System client documentation for the exact syntax and usage.

### Command summary for Sun PC-NFS users

You can use these commands with Sun PC-NFS:



-p	Causes a prompt for the user's MVS password. The password is passed to MVS to validate the user logging in. The type of security procedures required by the installation determines whether this parameter is required.
-n	Causes a prompt for a new password. The type of security procedures required by the installation determines whether this parameter is used.
-pn	Causes a prompt for the user's current password and then causes two prompts for the user's new password. The type of security procedures required by the installation determine whether this parameter is used.
-g <i>group</i>	A group name string passed to MVS for accounting purposes. The maximum length is 8 characters.
-a <i>account</i>	An account string passed to MVS for accounting purposes. The maximum length is 16 characters.
<i>hostname</i>	The name of the MVS host (for example, mvshost1).
<i>mvs_userid</i>	A user ID that MVS recognizes as valid. If you do not specify this parameter, your workstation user name is used. The z/OS NFS server does not support the use of an alias user ID or a mixed case user ID with the <b>mvslogin</b> command.



<i>drive</i>	Specifies the drive letter to which the mounted file system is attached.
<i>hostname</i>	The name of the MVS host (for example, mvshost1).
<i>mvs_qual</i>	The name of the high-level qualifier of the MVS files that you are accessing in the MVS file system (such as mvspartner1 or mvspartner1.test). The high-level qualifier specified must already exist in the MVS catalog.
<i>attribute</i>	A z/OS Network File System server data set creation or processing attribute (such as text). See "Chapter 10. Attributes for the z/OS NFS server" on page 75. If you specify any attributes, make sure you enclose <i>mvs_qual</i> and the attributes in double quotation marks.

```

>> showattr [-t] hostname [drive:]

```

<i>-t</i>	Used to specify tersed output.
<i>hostname</i>	The name of the MVS host (for example, mvshost1).
<i>drive</i>	Specifies the drive letter to which the mounted file system is attached.

```

>> net use drive: /d

```

<i>drive</i>	Specifies the drive letter to which the mounted file system is attached.
--------------	--

```

>> mvslogut hostname

```

<i>hostname</i>	The name of the MVS host (for example, mvshost1).
-----------------	---

## Mvslogin example

Suppose your MVS user ID is smith, and the name of the MVS system that you want to log onto is mvshost1. You would enter this command:

```
C> mvslogin mvshost1 smith
```

After you have logged on, you also need to issue a **net use** command before you can access MVS files.

## Net use example

Suppose the name of the MVS system that you want to log onto is mvshost1, and the high-level qualifier of the files that you want to access or create is test. If you are using PC-NFS and you want to use your Z: drive to mount the MVS files, you would enter this command:

```
C> net use z: mvshost1:test
```

After you have mounted an MVS file or directory, use standard DOS commands to access or create MVS files as if they were local files.

Suppose the default for the processing mode is binary, but you want to mount the test files using the text processing mode. You can do this with PC-NFS by entering this command:

```
C> net use z: mvshost1:"test,text"
```

## Showattr example

Suppose you want to see the values of the file attributes associated with the z/OS Network File System server. To do this, you could enter this command:

```
C> showattr mvshost1 z:
```

where `mvshost1` is the name of the MVS system, and `Z:` is the drive that you used to mount the MVS files. The data set creation attributes, processing attributes, and site attributes are all displayed. You can modify the data set creation attributes and the processing attributes using the **net use** command, but only the MVS system administrator can change the values of the site attributes.

## Net use /d example

Suppose you are finished working with a set of MVS files that you had mounted to your `Z:` drive, and you are ready to unmount them. You would enter this command:

```
C> net use z: /d
```

## Mvslogut example

Use the **mvslogut** command when you finish accessing files that reside on a specific MVS system or host. Invoking the **mvslogut** command prevents other users on your workstation from having access to files on that specific MVS system. You should issue the **mvslogut** command at the close of each session.

For example, if the name of the MVS system that you had logged onto is `mvshost1`, you would log off the MVS system by entering this command:

```
C> mvslogut mvshost1
```



---

## Chapter 8. Commands and examples for z/OS NFS clients

This chapter gives the syntax and examples of commands that z/OS NFS users need to know to access MVS, AIX, UNIX, z/OS, and other remote files using the z/OS NFS client. This chapter shows how to perform the following tasks:

- Log on to a remote MVS system from the z/OS NFS client if the target server is a remote z/OS NFS server.
- Access NFS files from the z/OS NFS client.
- Display z/OS NFS client statistical data.
- Query mount points.
- Display default mount point attributes.
- Mount and unmount remote file systems from the z/OS NFS client.
- Log out of MVS, if the target server was a remote z/OS NFS server.

The command programs are intended to run in an z/OS shell environment and are not implemented as TSO commands, with the exception of **mount** and **unmount**.

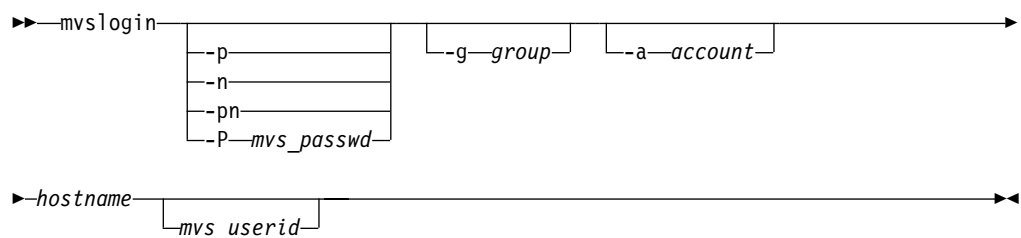
The **mount** and **unmount** commands are not part of z/OS NFS. See *z/OS UNIX System Services Command Reference* for additional details. An example of the syntax and usage is shown here for your information. You can use the **tso help mount** and **tso help unmount** commands to see the syntax that is applicable to your system.

For information about the z/OS NFS attributes see “Chapter 9. Attributes for the z/OS NFS client” on page 71 and “Chapter 10. Attributes for the z/OS NFS server” on page 75.

---

### Using commands on DFSMS/MVS

Here is a summary of the syntax for the commands described in this chapter (the **mount** and **unmount** commands are discussed later in this chapter):



-p	Causes a prompt for the user's MVS password. The password is passed to MVS to validate the user logging in. Your security procedures determine whether you should use this parameter.
-n	Causes a prompt for a new password.
-pn	Causes a prompt for the user's current password and then causes two prompts for the user's new password.
-P mvs_passwd	No prompt for your MVS password; just type your MVS password after the -P. This enables you to automate your MVS login.
-g group	A group name string passed to MVS for accounting purposes. The maximum length is 8 characters.

<i>-a account</i>	An account string passed to MVS for accounting purposes. The maximum length is 16 characters.
<i>hostname</i>	The name of the MVS host (for example, mvshost1). The default is the local host.
<i>mvs_userid</i>	A user ID that MVS recognizes as valid. If you do not specify this parameter, your workstation user name is used. The z/OS NFS server does not support the use of an alias user ID or a mixed case user ID with the <b>mvslogin</b> command.

➤➤ showattr -t *hostname* /localpath ➤➤

<i>-t</i>	Used to specify tersed output.
<i>hostname</i>	The name of the MVS host (for example, mvshost1). The default is the local host.
<i>/localpath</i>	The mount point on your client system (for example, /u/smith/mnt). This should be an empty directory.

➤➤ nfsstat 
 -c  
 -n  
 -r  
 -z  
 -m
  
 mount point
  ➤➤

<i>-p</i>	Causes a prompt for the user's MVS password. The password is passed to MVS to validate the user logging in. Your security procedures determine whether you should use this parameter.
<i>-c</i>	Displays both NFS and RPC statistics about the z/OS NFS client. This is the default option on the <b>nfsstat</b> command.
<i>-n</i>	Displays NFS statistics about the z/OS NFS client.
<i>-r</i>	Displays RPC statistics about the z/OS NFS client.
<i>-z</i>	Initializes statistics to zero. Used by root user only. This option can be combined with options <b>-c</b> , <b>-n</b> , and <b>-r</b> on the <b>nfsstat</b> command. When combined with these nfsstat options, each particular set of statistics is set to zero after the statistics are printed.
<i>-m</i>	Displays the name of each NFS mounted file system.
<i>-m mount point</i>	Displays information for the NFS mounted file system on the specified mount point.

➤➤ showmount 
 -a  
 -d  
 -e
  *hostname* ➤➤

If you omit the options, the default option displays hostnames of all remote mounts from the *hostname* NFS server.

-a	Displays all mounts in the format hostname:directory from the hostname specified in the <b>showmount</b> command.
-d	Displays only directory names of all mounts from the hostname specified in the <b>showmount</b> command.
-e	Displays the list of exported directories from the hostname specified in the <b>showmount</b> command.
hostname	The name of the NFS server host (for example, mvshost1). The default is the local host.

►►—os22mvs—*input*—*output*—◄◄

<i>input</i>	Absolute path name of the input file to be converted.
<i>output</i>	Absolute path name of the output file.

►►—mvs2os2—*input*—*output*—◄◄

<i>input</i>	Absolute path name of the input file to be converted.
<i>output</i>	Absolute path name of the output file.

►►—mvslogout—*hostname*—◄◄

hostname	The name of the MVS host (for example, mvshost1). The default is the local host.
----------	--

## Accessing MVS

To access remote MVS files, enter both the **mvslogin** command to log into the z/OS NFS server's host system and the **mount** command to mount the files or data sets to your local system. The **mvslogin** command is only required when accessing data on systems where the z/OS NFS server site *security* attribute is set to saf or safexp. Once the files or data sets are mounted to a local mount point, you can read, write, create, delete and treat the mounted files as part of your local z/OS UNIX files. When you are finished with your work, use the **unmount** and **mvslogout** commands to break the connection. The **mvslogout** command is only required when the **mvslogin** command was used to begin the connection.

**Note:** Issuing the **mount** and **unmount** commands, as well as creation of the mount point, must be performed by someone with superuser authority.

To access files on MVS systems where the z/OS NFS server site *security* attribute is set to saf, exports, or safexp, you need an MVS user ID and password, and authorization to access the files that you need. You can only establish one MVS session for each MVS user ID. If you do not already have an MVS user ID, an MVS password, and access authorization on the MVS system from which you require NFS services, request them from your MVS system administrator.

**Note:** If you cannot use the **mvslogin**, **mvslogout**, or **showattr** commands, they might be installed incorrectly. Ensure that your system administrator has made the

executable code for these three commands available to your MVS user ID and that you have been given the correct access authority to them.

## Mvslogin command examples

Use the **mvslogin** command to log in to the remote MVS system. The **mvslogin** command can be issued multiple times and the last one overrides the previous one. The **mvslogin** command is only required when accessing data on systems where the z/OS NFS server site *security* attribute is set to saf or safexp.

Table 16 shows examples of the **mvslogin** command where mvshost1 is the name of the MVS host and smith is the user's ID on MVS.

Table 16. Examples of the mvslogin command for clients

### Command Examples

mvslogin -p mvshost1 smith
mvslogin -P smithspw -g finance -a 5278 mvshost1 smith
mvslogin -n mvshost1 smith
mvslogin -pn -a 5278 mvshost1 smith
mvslogin mvshost1
mvslogin mvshost1 smith

In the example where the user enters **mvslogin mvshost1**, the current login client user ID is used as the MVS user ID.

In the example where the user enters **mvslogin mvshost1 smith**, the system then prompts for Smith's MVS password. If Smith logs in successfully, this message appears:

GFSA955I smith logged in ok.

Otherwise, an appropriate error message appears.

**Note:** Messages that start with GFSA and GFSC apply to NFS requests. See "Appendix A. Network File System messages" on page 93.

### "Permission denied" message

If you have successfully logged in and get the "Permission denied" message while trying to access data, that can be due to one of these cases:

- An **mvslogout** command for the same MVS host has been entered from the same client platform. See **mvslogout** for details.
- Your MVS user ID has been automatically logged out because the **logout** attribute value has been exceeded. This can happen when you leave the client idle for too long. Enter the **mvslogin** command again, and start your processes again. To find out how many seconds you can stay logged in while your client is idle, issue the **showattr** command and look at the **logout** attribute.
- Another mvslogin to the same MVS host using the same MVS ID has been entered from the same UID and the same client platform. If this is the case, retry your access.



# Mount command syntax and examples

Use the **tso mount** command to make a connection between a mount point on your local HFS file system and one or more files on a remote MVS, AIX, UNIX, z/OS, or other file system. The **mount** command can only be used by a MVS superuser.

**Note:** The same mount function can also be performed using the z/OS UNIX **automount** facility or **/etc/rc** shell scripts support. z/OS UNIX does not support NFS mounts in the SYS1.PARMLIB member statement. When the automount facility is used to manage remote NFS mount points, the z/OS NFS client user could experience ESTALE/EIO errors if the automounter unmounts the accessed mount point when the time limits specified by the automount DURATION and DELAY parameters have been exceeded. The automount default, DURATION=NOLIMIT, disables the DURATION timeout period. The DELAY for unmounting file systems should be longer than the time z/OS NFS clients are likely to keep NFS mounts to the files and directories active. For more information see *z/OS UNIX System Services File System Interface Reference*.

The same mount function can also be performed using the z/OS UNIX **automount** facility or **/etc/rc** shell scripts support. z/OS UNIX does not support NFS mounts in the SYS1.PARMLIB member statement. When the automount facility is used to manage remote NFS mount points, the z/OS NFS client user could experience ESTALE/EIO errors if the automounter unmounts the accessed mount point when the time limits specified by the automount DURATION and DELAY parameters have been exceeded. The automount default, DURATION=NOLIMIT, disables the DURATION timeout period. The DELAY for unmounting file systems should be longer than the time z/OS NFS clients are likely to keep NFS mounts to the files and directories active. For additional information see *z/OS UNIX System Services File System Interface Reference*.

Figure 7 illustrates the syntax of the **mount** command:

```
MOUNT FILESYSTEM(file_system_name)
      TYPE(NFS)
      MOUNTPOINT(local_mountpoint)
      MODE(RDWR|READ)
      PARM('hostname:"path_name,server_attributes", options')
      SETUID|NOSETUID
      WAIT|NOWAIT
```

Figure 7. TSO mount command syntax operands

Table 17. Mount operand

Operand	Description
<b>FILESYSTEM(nfile_system_name)</b>	Specifies the name of the file system to be added to the file system hierarchy. This operand is required. The file system name specified must be unique among previously mounted file systems. It may be an arbitrary name up to 44 characters in length of a filesystem. You can enclose file_system_name in single quotes, but they are not required.
<b>TYPE(NFS)</b>	Specifies the type of file system that performs the logical mount request. The NFS parameter must be used.

Table 17. Mount operand (continued)

Operand	Description
<b>MOUNTPPOINT</b> ( <i>local_mountpoint</i> )	<p>Specifies the path name of the mount point directory, the place within the file hierarchy where the file system is to be mounted. The <i>local_mountpoint</i> specifies the mount point path name. The name can be a relative path name or an absolute path name. The relative path name is relative to the current working directory of the TSO session (usually the home directory). Therefore, you should usually specify an absolute path name. A path name is case-sensitive, so enter it exactly as it is to appear. Enclose the path name in single quotes.</p> <p><b>Note:</b> The mount point must be a directory. Any files in that directory are inaccessible while the file system is mounted. Only one file system can be mounted to a mount point at any time.</p>
<b>MODE</b> ( <i>RDWR READ</i> )	<p>Specifies the type of access for which the file system is to be opened.</p> <p><b>RDWR</b> specifies that the file system is to be mounted for read and write access. <b>RDWR</b> is the default option.</p> <p><b>READ</b> specifies that the file system is to be mounted for read-only access.</p>
<b>PARM</b> ('hostname:"path_name,server_attributes", options')	<p>Specifies the hostname of the remote Network File System server, the server attributes, and the options. The double quotes are omitted if no server attributes are specified. Enclose the entire string in single quotes. You can specify lowercase or uppercase characters.</p>
<b>SETUID NOSETUID</b>	<p>See <i>z/OS UNIX System Services Command Reference</i> for details about <b>TAGSETUID</b> and <b>NOSETUID</b>.</p>
<b>TAG</b> ( <i>TEXT,CCSID</i> )	<p>See <i>z/OS UNIX System Services Command Reference</i> for details about the <b>TAG</b> keyword.</p> <p><b>Note:</b> When <b>TAG</b> is specified, <b>xlat(Y)</b> must not be specified; otherwise, mount will fail.</p>
<b>WAIT NOWAIT</b>	<p>See <i>z/OS UNIX System Services Command Reference</i> for details on <b>WAIT</b> and <b>NOWAIT</b>.</p>

## Data conversion

The z/OS NFS client supports data conversion defined by the universal character encoding standard known as the Unicode Standard on z/OS V1R2 (and above) when reading data from a remote NFS server or writing data to a remote z/OS NFS server. The Unicode Standard offers character conversion as well as basic case conversion. Within character conversion, characters are converted from one coded character set identifier (CCSID) to another. CCSID information is obtained from the **cln\_ccsid** and **srv\_ccsid** parameters.

On z/OS releases below V1R2, the z/OS NFS client supports data conversion defined by the Character Data Representation Architecture (CDRA) when reading data from a remote NFS server or writing data to a remote NFS server. CDRA characters are converted from one CCSID to another. CCSID information is obtained from the **cln\_ccsid** and **srv\_ccsid** parameters. See *Character Data Representation Architecture Reference and Registry* and *Character Data Representation Architecture Overview* for additional information.

Only single byte to single byte data conversion is supported. For example, if a client file has a CCSID of 437 and a server file has a CCSID of 297, data conversion will occur between USA ASCII format (CCSID 437) and French EBCDIC format (CCSID 297). Single byte to multiple byte conversion (including double byte character set (DBCS)) is not supported and will result in NFS terminating with an error message.

The **cln\_ccsid**, **srv\_ccsid**, and **xlat** parameters identify whether data conversion is performed, and how data conversion is done. The **cln\_ccsid**, **srv\_ccsid**, and **xlat** parameters are supported by the z/OS NFS client installation parameter and **tso mount** command. The parameters on a **tso mount** command override the parameters specified as a z/OS NFS client installation parameter.

### **BSAM, QSAM, and VSAM ESDS access to remote files**

BSAM, QSAM, and VSAM ESDS applications can access files stored on remote NFS servers through the z/OS NFS client. This will allow existing MVS application programs access to data on other systems using BSAM, QSAM, and VSAM ESDS interfaces. The BSAM, QSAM, and VSAM ESDS access methods assume that all text files are EBCDIC. When using these access methods, the **delim** parameter indicates whether the remote files contain text or binary data. Text data consists of records that are separated by a delimiter. If the **delim** parameter is not binary, the EBCDIC text delimiter is used by the access methods when processing the remote files. The **delim** parameter is supported on the z/OS NFS client installation parameter and **tso mount** command.

All the remote file objects under the same mount point have the same **delim** value. The **delim** parameter cannot be set on a file basis under the mount point. The **delim** parameter in the **tso mount** command overrides the **delim** parameter specified in z/OS NFS client installation parameter. However, you can override the **delim** parameter on the **tso mount** command with the **filedata** parameter on a JCL DD statement, SVC 99, or **tso allocate** command. The **filedata** parameter can be either text or binary.

For BSAM, QSAM, and VSAM ESDS applications accessing files stored on remote NFS servers, the z/OS NFS client provides data conversion when the **xlat=Y** parameter is specified according to the **cln\_ccsid** and **srv\_ccsid** settings. When **xlat=N**, the z/OS NFS client will not perform data conversion. The **filedata** parameter on a JCL DD statement is also used to specify if the data consists of text records separated by delimiters or if the data is binary and does not contain record delimiters. To avoid undesirable data conversions, care should be taken to insure the specification of the **xlat** and **delim** parameters are not in conflict with the data type specified by the **filedata** parameter on a JCL DD statement. The **filedata** and **delim** parameters only affect BSAM, QSAM, and VSAM ESDS data access and have no effect on the z/OS NFS client data conversion. The z/OS NFS client data conversion is only controlled by the **xlat** parameter. The significance of different **filedata** and **delim** combinations are as follows:

**Note:** In each case insure the z/OS NFS client **xlat=Y**, **cln\_ccsid**, and **srv\_ccsid** parameter settings are correct for the **filedata** and **delim** combination.

---

#### **FILEDATA=TEXT, Delim=notBINARY**

Means that the data is to be accessed as text. The access method appends a record delimiter on output and expects delimiters on input. The delimiter used is that specified on the **delim** parameter.

---

#### **FILEDATA=TEXT, Delim=BINARY**

Means that the data is to be accessed as text. The access method appends a record delimiter on output and expects delimiters on input. The delimiter used is the default of the EBCDIC new line character (x'15') since the **delim** parameter does not specify a valid text delimiter.

---

#### **FILEDATA=BINARY, Delim=notBINARY**

Means that the data is to be accessed as binary. The access method does not append record delimiters on output, does not recognize record delimiters on input, and it treats all characters as data on input.

---

---

**FILEDATA=BINARY, Delim=BINARY**

Means that the data is to be accessed as binary. The access method does not append record delimiters on output, does not recognize record delimiters on input, and it treats all characters as data on input.

---

**FILEDATA=not specified, Delim=specified**

Means that the data is to be accessed according to the value specified in the **delim** parameter.

---

**FILEDATA=not specified, Delim=not specified**

Means that the data is to be accessed as binary. The access method does not append record delimiters on output, does not recognize record delimiters on input, and it treats all characters as data on input.

---

The z/OS NFS client also provides UNIX authentication for security and provides the z/OS UNIX client's user ID (UID), group ID (GID), and a list of users GIDs to the remote NFS server for authorization checking. When the remote NFS server is the z/OS NFS server, the **mvslogin** command can be used to provide additional security checking through RACF authentication. MVS application programs which require access to data on remote z/OS systems may be required to perform an **mvslogin**.

For information on BSAM, QSAM, and VSAM ESDS applications access to HFS or remote files and their restrictions, refer to *z/OS DFSMS: Using Data Sets*.

### Mount command examples

Use the **tso mount** command to make a connection between a local mount point on the z/OS NFS client and an NFS server.

In this example, the **mount** command is used to mount a set of MVS files. The **PARM** operand contains the z/OS NFS server *text* processing attribute and requires the use of double quotes around the string *user,text*.

```
mount filesystem(nfs00) type(nfs) mountpoint('/u/nfsdir')
parm('stlmvs3:"user,text",soft,timeo(100)')
```

nfs00	The name of the file system to be added to the file system hierarchy
nfs	The required file system type
/u/nfsdir	The name of the mount point (preferably an empty directory)
stlmvs3	The name of the host z/OS NFS server
user	The name of the high-level qualifier of the MVS files on the z/OS NFS server
text	Processing attribute for the z/OS NFS server
soft	PARM operand option for the z/OS NFS client
timeo(100)	PARM operand option for the z/OS NFS client

In this example, the **mount** command is used to mount a MVS HFS directory. The **PARM** operand contains the z/OS NFS server *binary* processing attribute and requires the use of double quotes around the string */hfs/u/user,binary*.

```
mount filesystem(nfs01) type(nfs) mountpoint('/u/nfsdir1')
parm('stlmvs3: "/hfs/u/user,binary",soft')
```

nfs01	The name of the file system to be added to the file system hierarchy
-------	--

nfs	The required file system type
/u/nfsdir1	The name of the mount point (preferably an empty directory)
stlmvs3	The name of the host z/OS NFS server
/hfs/u/user	The name of the HFS directory on the z/OS NFS server
binary	Processing attribute for the z/OS NFS server
soft	PARM operand option for the z/OS NFS client

In this example, the **mount** command is used to mount an AIX home directory.

```
mount filesystem(nfs02) type(nfs) mountpoint('/u/nfsdir2')
parm('aix6000:/home/user,xlat(y)')
```

nfs02	The name of the file system to be added to the file system hierarchy
nfs	The required file system type
/u/nfsdir2	The name of the mount point (preferably an empty directory)
aix6000	The name of the host AIX NFS server
/home/user	The name of the home directory on the AIX NFS server
xlat(y)	PARM operand option for the z/OS NFS client

In this example, the **mount** command is used to mount an AIX home directory using the NFS version 3 protocol.

```
mount filesystem(nfs03) type(nfs) mountpoint('/u/nfsdir2')
parm('aix6000:/home/user,xlat(y),vers(3)')
```

nfs03	The name of the file system to be added to the file system hierarchy
nfs	The required file system type
/u/nfsdir2	The name of the mount point (preferably an empty directory)
aix6000	The name of the host AIX NFS server
/home/user	The name of the home directory on the AIX NFS server
xlat(y)	PARM operand option for the z/OS NFS client
vers(3)	The version of NFS protocol that is being used

## Getting authorization to access files

If the mount fails, check with your system administrator to ensure that you are authorized to access the HFS, AIX, or UNIX file system listed in the exports control file. The z/OS NFS client also provides UNIX authentication for security and provides the z/OS UNIX client's UID, GID, and a list of the GIDs from the z/OS UNIX client's groups to the remote NFS server for authorization checking. When the remote NFS server is the z/OS NFS server, the **mvslogin** command can be used to provide additional security checking through RACF authentication. The privilege level required to enter **mount** and **unmount** commands is superuser. When requesting service from the z/OS NFS server, if the MVS system operator issues the *freeze=on* operand of the **modify** command, all new tries to mount an HFS filesystem fail until the MVS system operator issues the *freeze=off* operand. If the z/OS UNIX system operator issues the *freeze=onhfs* operand of the **modify**

command, conventional MVS data sets can still be mounted, but all new tries to mount z/OS UNIX files fail until the system operator issues the *freeze=offhfs* operand.

### **Saving of mount points**

Once the **mount** command is issued successfully and a mount point is established between a remote directory and the HFS, the mount point information is not saved by the z/OS NFS client. The z/OS NFS client does not maintain mount persistence across restart. If z/OS UNIX or the z/OS NFS client is restarted, all prior session's mount point information is lost and all mount points must be reestablished.

### **Automatic timed logout - logout attribute**

When using Network File System services from the z/OS NFS server, if there is no activity on the client within the period specified in the **logout** attribute of the attributes file, or if the server stops, the connection between the server and the client workstation is logged out automatically. You must issue the **mvslogin** command again to get access to the MVS files.

## **Displaying client statistical information—nfsstat**

Use the **nfsstat** command to display the z/OS NFS client statistical information, to reset the statistical information to zero, to display NFS mount point information, or to set the debug status.

The following **nfsstat** command displays the NFS and RPC statistics for the z/OS NFS client:

```
nfsstat -c
```

The following **nfsstat** command initializes the NFS and RPC statistics for the z/OS NFS client to zero. This option may be used by the root user only:

```
nfsstat -z
```

Figure 8 on page 63 shows the output from the **nfsstat** command using the **-c** option to display the RPC and NFS statistics for the NFS client.

```

USER1:/u/user1:>nfsstat -c

Client rpc:
calls          badcalls      retrans        timeout
51             0              0              0

Client nfs_V2:
calls          badcalls
45             0
null           getattr      setattr        root          lookup
0             0% 21 47% 1 2% 0 0% 13 29%
readlink       read         writecache     write         create
0             0% 0 0% 0 0% 0 0% 1 2%
remove         rename       link           symlink       mkdir
4             9% 0 0% 0 0% 0 0% 0 0%
rmdir          readdir      fsstat
0             0% 3 7% 2 4%

Client nfs_V3:
calls          badcalls
4             0
null           getattr      setattr        lookup        access
0             0% 0 0% 0 0% 0 0% 0 0%
readlink       read         write          create        mkdir
0             0% 0 0% 0 0% 0 0% 0 0%
symlink        mknod        remove         rmdir         rename
0             0% 0 0% 0 0% 0 0% 0 0%
link           readdir      readdirplus    fsstat        fsinfo
0             0% 0 0% 0 0% 2 50% 1 25%
pathconf       commit
1             25% 0 0%

```

Figure 8. Displaying NFS client rpc and NFS statistical information

Figure 9 shows the output from the **nfsstat** command using the -r option to display the remote procedure call (RPC) statistics for the NFS client.

```

#nfsstat -r
Client rpc:
calls          badcalls      retrans        timeout
61             0              20005          20005

```

Figure 9. Displaying NFS client rpc statistical information

calls	Total number of RPC calls sent
badcalls	Total of RPC calls rejected by a server
retrans	Number of times an RPC call had to be retransmitted
timeout	Number of times an RPC call timed out

Figure 10 on page 64 shows the output from the **nfsstat** command using the -n option to display the NFS statistics for the NFS client.

```
# nfsstat -n

Client nfs_V2:
calls      badcalls
45          0
null       getattr    setattr    root       lookup
0           0% 21    47% 1         2% 0         0% 13      29%
readlink   read        writecache write       create
0           0% 0     0% 0         0% 0         0% 1        2%
remove     rename      link       symlink     mkdir
4           9% 0     0% 0         0% 0         0% 0        0%
rmdir      readdir    fsstat
0           0% 3     7% 2         4%

Client nfs_V3:
calls      badcalls
4           0
null       getattr    setattr    lookup     access
0           0% 0     0% 0         0% 0         0% 0        0%
readlink   read       write      create     mkdir
0           0% 0     0% 0         0% 0         0% 0        0%
symlink    mknod      remove     rmdir      rename
0           0% 0     0% 0         0% 0         0% 0        0%
link       readdir    readdirplus fsstat     fsinfo
0           0% 0     0% 0         0% 2         50% 1       25%
pathconf   commit
1           25% 0     0%
```

Figure 10. Displaying NFS client NFS statistical information

calls	Total number of NFS calls sent
badcalls	Total of NFS calls rejected by a server

Figure 11 shows the output from the **nfsstat** command using the -m option to display the server and path name of each NFS mounted file system.

```
#nfsstat -m
mvshost:/sj/sjpl is mounted on /sj/sjpl/host1
filesystem NFS_MNT1
mvshost2:/sj/sjpl2,text mounted on /sj/sjpl/host2
filesystem NFS_MNT2
```

Figure 11. Displaying NFS mounted file system information

Figure 12 on page 65 and Figure 13 on page 65 show the output from the **nfsstat** command using the -m option to display the server name, path name, and attributes of mount point */sj/sjpl/host1* from both version 2 and version 3 protocols.



```
#nfsstat -m /sj/sjpl/host1
server mvshost
path /sj/sjpl/host1

hard vers(2) retry(10) timeo(7) retrans(3)
delim(binary) xlat(n) cln_ccsid(1047) srvccsid(819)
rsize(8192) wsize(8192) readahead(8) delaywrite(16)
acregmin(3) acregmax(60) accdirmin(30) accdirmax(60)
datacaching(y) attrcaching(y) dynamicsizeadj(y)
```

Figure 12. Displaying V.2 protocol NFS mounted file system information

```
#nfsstat -m /sj/sjpl/host1
server mvshost
path /sj/sjpl/host1

hard vers(3) retry(10) timeo(7) retrans(3)
delim(binary) xlat(n) cln_ccsid(1047) srvccsid(819)
rsize(32768) wsize(32768) readahead(8) delaywrite(16)
acregmin(3) acregmax(60) accdirmin(30) accdirmax(60)
datacaching(y) attrcaching(y) dynamicsizeadj(y)
```

Figure 13. Displaying version 3 protocol NFS mounted file system information

## Displaying server mount information—showmount

Use the **showmount** command to display the remote NFS server mount information. If you omit the options, the default option displays hostnames of all remote mounts from the hostname NFS server. If you omit the *hostname* parameter, then the local hostname is used.

The following **showmount** command displays all remote mounts in the format *hostname:directory* from the local *hostname* NFS server:

```
showmount -a
```

The following **showmount** command displays only the directory names of all the remote mounts from the local *hostname* NFS server:

```
showmount -d
```

Figure 14 shows the output from the **showmount** command using the **-a** option to display all mounts in the format *hostname:directory* from the hostname *mvshost*.

```
# showmount -a mvshost
mvshost.sanjose.ibm.com:/IBMUSER
usera.sanjose.ibm.com:/USER2
```

Figure 14. Displaying server mount hostname and directory information

Figure 15 shows the output from the **showmount** command using the -d option to display only the directory names of all mounts from the hostname mvshost.

```
# showmount -d mvshost
/IBMUSER
/USER2
```

*Figure 15. Displaying server mount directory information*

Figure 16 shows the output from the **showmount** command with no option specified to display only the hostnames of all remote mounts from the hostname mvshost.

```
# showmount mvshost
mvshost.sanjose.ibm.com
usera.sanjose.ibm.com
```

*Figure 16. Displaying server mount hostname information*

Figure 17 shows the output from the **showmount** command using the -e option to display the exported directories from the hostname aix\_server1.

```
USER1:/u/user1:>showmount -e aix_server1
Export list for host aix_server1:
/home/u/guest/test (everyone)
/usr/lpp/info      (everyone)
/tmp               (everyone)
```

*Figure 17. Displaying server export information*

Figure 18 shows the output from the **showmount** command using the -e option to display the exported directories from the hostname mvshost. In this case, mvshost has the site attribute set to **security(none)**.

```
# showmount -e mvshost
No exported file systems for host MVSHOST
```

*Figure 18. Displaying server export information, security(none)*

Figure 19 on page 67 shows the output from the **showmount** command using the -e option to display the exported directories from the hostname mvshost. In this case, mvshost has the site attribute set to **security(safexp)**.

```
# showmount -e mvshost
Export list for host MVSHOST:
/IBMUSER          user1
```

Figure 19. Displaying server export information, security(*safexp*)

## Displaying default and mount point attributes—**showattr**

Use the **showattr** command to display the default attributes or the attributes that have been set for a specific mount point of the NFS server. If you specify a mount point, **showattr** shows the attributes for the mount point, including the overriding values. For descriptions of the attributes, see “Chapter 10. Attributes for the z/OS NFS server” on page 75 and “Mount command syntax and examples” on page 57.

If you omit the *hostname*, you must specify the */localpath*.

The following is an example of the **showattr** command:

```
showattr mvshost1 /u/smith/mnt
```

Make sure that your version of the **showattr** command matches the release of the z/OS NFS that you are using. Otherwise, the z/OS NFS server attributes will not display.

These examples show different ways you can use the **showattr** command.

Figure 20 on page 68 shows a **showattr** command with just the hostname (*mvshost1* in this example) specified. The default attributes for the server are displayed.

```

USER1:/u/user1:>showattr mvshost1

z/OS Network File System Server Data Set Creation Attributes:

lrecl(8196)          recfm(vb)          blksize(0)
space(100,10)        blks              dsorg(ps)
dir(27)              unit()            volume()
recordsize(512,4K)   keys(64,0)        nonspanned
shareoptions(1,3)
mgmtclas()           dsntype(pds)       norlse
dataclas()           storclas()

z/OS Network File System Server Processing Attributes:

binary              lf              blankstrip
nofastfilesize      noretrieve     maplower
mapleaddot          executebitoff   setownerroot
attrtimeout(120)    readtimeout(90) writetimeout(30)
sync               nofileextmap    xlat(OEMVS)
srv_ccsid(0)        cln_ccsid(0)    nordrverf
sidefile(hlq.nfs.mapping)

z/OS Network File System Server Site Attributes (not modifiable):

mintimeout(1)       nomaxtimeout     logout(1800)
nfstasks(8,8)       restimeout(48,0) hfs(/hfs)
bufhigh(32M)        readaheadmax(16K) cachewindow(128)
percentsteal(20)    maxrdforszleft(32) logicalcache(16M)
smf(userfile)       pcnfsd           security(exports,exports,exports)
leadswitch          sfmax(0)         nochecklist
fn_delimiter(,)     readdirttimeout(30)
public(NFSUSR3,/HFS/usr/lpp/internet/server_root/pub)

```

Figure 20. Displaying default attributes

If you use the terse (-t) option, the attributes display like this:

```

USER1:/u/user1:>showattr -t mvshost1

lrecl(8196),recfm(vb),blksize(0),space(100,10),blks,dsorg(ps),dir(27),unit(),
volume(),recordsize(512,4K),keys(64,0),nonspanned,shareoptions(1,3),mgmtclas(),
dsntype(pds),norlse,dataclas(),storclas()
binary,lf,blankstrip,nofastfilesize,retrieve,maplower,mapleaddot,executebitoff,
setownerroot,attrtimeout(120),readtimeout(90),writetimeout(30),sync,
nofileextmap,xlat(oemvs),srv_ccsid(0),cln_ccsid(0),nordrverf,sidefile(hlq.nfs.mapping)
mintimeout(1),nomaxtimeout,logout(1800),nfstasks(8,8),restimeout(48,0),
hfs(/hfs),bufhigh(32M),readaheadmax(16K),cachewindow(128),percentsteal(20),
maxrdforszleft(32),logicalcache(16M),smf(userfile),pcnfsd,
security(exports,exports,exports),leadswitch,sfmax(0),nochecklist,
fn_delimiter(,),readdirttimeout(30),
public(NFSUSR3,/HFS/usr/lpp/internet/server_root/pub)

```

## Unmounting and logging out of MVS

This section describes the **unmount** and **mvslogout** commands.

### Disconnecting your mount point - unmount

Use the **unmount** command to break the connection between the mount point on your client and the server (that is, to unmount). You must have superuser authority to issue the unmount command.

**Note:** The same unmount function can also be performed using the z/OS UNIX **automount** facility or /etc/rc shell scripts support. When the **automount** facility is used to manage remote NFS mount points, the z/OS NFS client user could experience ESTALE/EIO errors if the automounter unmounts the accessed mount

point when the time limits specified by the **automount** *duration* and *delay* parameters have been exceeded. For additional information on the z/OS UNIX **automount** facility (*/etc/rc* shell scripts support) see *z/OS UNIX System Services Planning* and *z/OS UNIX System Services File System Interface Reference*.

Figure 21 illustrates the syntax of the **unmount** command.

```
UNMOUNT FILESYSTEM(file_system_name)
NORMAL | DRAIN | IMMEDIATE | FORCE | RESET
```

Figure 21. TSO unmount command syntax operands

---

**FILESYSTEM(file\_system\_name)**

Specifies the name of the file system to be removed from the file system hierarchy. *file\_system\_name* specifies the *file\_system\_name* exactly as it was specified when the file system was originally mounted. You can enclose *file\_system\_name* in single quotes, but they are not required.

---

**NORMAL|DRAIN|IMMEDIATE|FORCE|RESET**

**NORMAL:** Specifies that if no user is accessing any of the files in the specified file system, the unmount request is processed. Otherwise, the system rejects the unmount request. **NORMAL** is the default option.

**DRAIN:** Specifies that the system is to wait until all uses of the file system have ended normally before the unmount request is processed or until another unmount command is issued.

**Note:** **unmount** can be specified with **IMMEDIATE** to override a previous **unmount DRAIN** request for a file system. If this is used in the foreground, your TSO session waits until the **unmount** request has completed. The attention request key (usually ATTN or PA1) will not end the command.

**IMMEDIATE:** Specifies that the system is to unmount the file system immediately. Any users accessing files in the specified system receive failing return codes. All data changes to files in the specified file system are saved. If the data changes to files cannot be saved, the unmount request fails.

**FORCE:** Specifies that the system is to unmount the file system immediately. Any users accessing files in the specified file system receive failing return codes. All data changes to files in the specified file are saved, if possible. If the data changes cannot be saved to the files, the unmount request continues and data is lost.

**Note:** You must issue an **unmount IMMEDIATE** request before issuing **unmount FORCE**. Otherwise, **unmount FORCE** fails.

**RESET:** A reset request stops a previous **unmount DRAIN** request.

---

The following example unmounts the file system NFSC\_001 normally:

```
UNMOUNT FILESYSTEM('NFSC_001')
```

The following example forces an unmount of the file system NFSC\_001. You must issue an **unmount IMMEDIATE** before you can issue an **unmount FORCE** command.

```
UNMOUNT FILESYSTEM('NFSC_001') IMMEDIATE
UNMOUNT FILESYSTEM('NFSC_001') FORCE
```

If you receive a “**No Such File or Directory**” message, the MVS system operator can also unmount your workstation from the server. If this happens before you try to unmount, you get a “No such file or directory” error message.

## Ending your MVS session - mvslogout

Use the **mvslogout** command to disconnect from the remote z/OS NFS server host. The **mvslogout** command is only required when the **mvslogin** command was used to begin the connection.

An **mvslogout** to an MVS user ID cancels a prior **mvslogin** to the same MVS user ID from the same local host.

Your account is automatically logged out if it is inactive for the period of time specified in the **logout** site attribute.

The following example disconnects the client from the remote z/OS NFS server machine, **mvshost1**:

```
mvslogout mvshost1
```

---

## Chapter 9. Attributes for the z/OS NFS client

This chapter contains information about the attributes that are used by the z/OS NFS client.

Table 18 contains directive information about this chapter's contents:

*Table 18. Attributes - z/OS NFS client*

Section	Page
Client attributes	71
DataCaching attribute	73
Mount processing parameters and installation parameters	74

---

### Client attributes

Client attributes are described in Table 19.

*Table 19. Client attributes*

Attribute	Description
<b>acdirmax(<i>n</i>)</b>	Specifies the maximum lifetime in seconds of cached file attributes. The default value is <u>60</u> .
<b>acdirmin(<i>n</i>)</b>	Specifies the minimum lifetime in seconds of cached file attributes. The default value is <u>30</u> .
<b>acregmax(<i>n</i>)</b>	Specifies the maximum lifetime in seconds of cached file attributes. The default value is <u>60</u> .
<b>acregmin(<i>n</i>)</b>	Specifies the minimum lifetime in seconds of cached file attributes. The default value is <u>3</u> .
<b>AttrCaching(<i>Y/N</i>)</b>	<p>Specifies whether to process attributes and data caching. The default value is <u>Y</u>.</p> <p>If attribute caching is in effect, the z/OS NFS client maintains cache consistency with the copy of the file on the NFS server by performing the consistency check with the cached file attributes. When a file's data is read, it remains valid on the z/OS NFS client until the attribute cache is timed out or negated. If <b>AttrCaching(N)</b> is specified, it will automatically set <b>DataCaching(N)</b>.</p>
<b>Biod(<i>n</i>)</b>	<p>Specifies the number of asynchronous block input/output (I/O) daemons. The valid range is 1 to 32, the default value is <u>6</u>.</p> <p>The <b>Biod</b> daemon runs on all NFS client systems. When a user on a client wants to read or write to a file on a server, the Biod daemon sends this request to the server. The <b>Biod</b> daemon is activated during system startup and runs continuously.</p> <p>The number of daemons is based on the load the client can handle. Six to eight daemons can handle an average load. You must run at least one daemon for NFS to work.</p>
<b>Bufhigh(<i>n</i>)</b>	Specifies the storage limit for the NFS client. The valid range is 1 MB to 128 MB. The default value is <u>32</u> MB.
<b>cln_ccsid(<i>x</i>)</b>	Specifies the Coded Character Set Identifier (CCSID) for the local mounted file system. The default is <u>1047</u> (LATIN OPEN SYSTEM EBCDIC).

Table 19. Client attributes (continued)

Attribute	Description
<b>DataCaching(Y/N)</b>	<p>Specifies whether to perform data caching. The default value is <u>Y</u>.</p> <p>The <b>DataCaching</b> attribute provides finer granularity in controlling whether file data should be cached by the z/OS NFS client. By caching the file data, all subsequent references to the cached data is done locally thus avoiding the network overhead. This has additional significance when obtaining data from Network File System server systems which do not use UNIX access permissions for security as there is a potential security exposure allowing unauthorized users to access file data.</p>
<b>DelayWrite(n)</b>	<p>Specifies the maximum number of disk blocks for delay write. The valid range is 0 to 32. The default value is <u>16</u>. The blocksize is 8192. This option is only valid when <b>DataCaching=Y</b>.</p>
<b>Delim(binary nl cr lf crlf lfcr)</b>	<p>Specifies the line delimiter for record access to remote file through basic sequential access method (BSAM), queued sequential access method (QSAM), and virtual storage access method (VSAM). The default value is <u>binary</u>.</p> <p><i>binary</i> specifies the data does not have record delimiters. The access method does not add a delimiter for each record on output and treats any delimiters on input as data.</p> <p>The following values specify text:</p> <p><i>cr</i> – Specifies that records are delimited by the EBCDIC carriage return character (x'0D').</p> <p><i>crlf</i> – Specifies that records are delimited by the EBCDIC carriage return character followed by the EBCDIC line feed character (x'0D25').</p> <p><i>crnl</i> – Specifies that records are delimited by the EBCDIC carriage return character followed by the EBCDIC new line character (x'0D15').</p> <p><i>lf</i> – Specifies that records are delimited by the EBCDIC line feed character (x'25').</p> <p><i>lfcr</i> – Specifies that records are delimited by the EBCDIC line feed character followed by the EBCDIC carriage return character (x'250D').</p> <p><i>nl</i> – Specifies that records are delimited by the EBCDIC new line character (x'15').</p>
<b>DynamicSizeAdj(Y/N)</b>	<p>Specifies whether to perform the packet size adjustment for remote procedure call (RPC). The default value is <u>Y</u>.</p>
<b>proto(tcp udp)</b>	<p>Specifies the transport protocol for the NFS client to communicate with the NFS server. If <b>proto(tcp)</b> is specified, the server does not support <i>tcp</i>; <i>udp</i> is used instead. By default, the NFS client selects the <b>proto</b> and <b>vers</b> with the following priorities:</p> <ol style="list-style-type: none"> <li>1. <i>proto(tcp)</i> and <i>vers(3)</i></li> <li>2. <i>proto(udp)</i> and <i>vers(3)</i></li> <li>3. <i>proto(tcp)</i> and <i>vers(2)</i></li> <li>4. <i>proto(udp)</i> and <i>vers(2)</i></li> </ol>
<b>ReadAhead(n)</b>	<p>Specifies the maximum number of disk blocks to read ahead. The block size is 8192 bytes. The valid range is 0 to 16. The default value is <u>1</u>.</p>
<b>retrans(n)</b>	<p>Sets the number of time to retransmit the NFS remote procedure calls (RPC). The valid range is 0 to 1000 and the default value is <u>3</u>. This option is only valid when <i>soft</i> is specified.</p>



Table 19. Client attributes (continued)

Attribute	Description
<b>retry(<i>n</i>)</b>	Specifies the number of time to retry the mount operation. The valid range is 0 to 20,000. The default value is <b>10000</b> . This option is only valid during mount processing.
<b>resize(<i>n</i>)</b>	Sets the read buffer size to <i>n</i> bytes. The valid range is 1 to 8192, and the default value is <b>8192</b> .
<b>soft hard</b>	<p>Returns an error if the NFS server does not respond or continue to retry the NFS remote procedure call (RPC) until the NFS server responds.</p> <p>If you specify <i>hard</i>, then continue to retry NFS remote procedure call (RPC) until the NFS server responds. If <i>soft</i> is specified, an error is returned if the NFS server does not respond. Specify the maximum number of retries with the <b>retrans</b> option. The default value is <b>hard</b>. This option is valid for all NFS RPCs under the mount point.</p>
<b>srv_ccsid(<i>x</i>)</b>	<p>Specifies the Coded Character Set Identifier (CCSID) for the remote mounted file system.</p> <p>The default is <b>819</b> (ISO 8859–1 ASCII).</p>
<b>timeo(<i>n</i>)</b>	Sets the remote procedure call (RPC) to <i>n</i> tenths of a second. The default value is <b>7</b> .
<b>vers(2 3)</b>	Specifies the NFS protocol version that the client will use to communicate with the NFS server. If no version is specified, the z/OS NFS client will communicate with the NFS server at the highest protocol level that is supported by the server.
<b>wsize(<i>n</i>)</b>	Sets the write buffer size to <i>n</i> bytes. The default value is <b>8192</b> .
<b>xlat(Y N)</b>	If Y is specified, the data in all the files are text and the NFS client will perform data conversion according to the <i>cln_ccsid</i> and <i>srv_ccsid</i> parameters. The default value is <b>N</b> and should be used for binary data.

## DataCaching attribute

Security checking is done on the Network File Systemserver to determine whether the requesting client user is authorized to access the data. On UNIX systems, this is done by validating the client's user ID and group ID against the file's permission codes. If the authorization checking is successful, the file data is returned to the z/OS NFS client system. Further authorization checking for subsequent access to the cached data or for other client users is done on the z/OS NFS client system.

For conventional MVS data set access through the z/OS NFS server, the user is required to present their MVS credentials which are checked by the MVS security system, such as RACF, before file data is returned. Since the MVS system does not maintain UNIX style permission codes for conventional data sets, the z/OS NFS server returns a code indicating that anyone can access the file. This is done since passing any lesser access code to the client would result in the client user not being allowed to use the cached data which was already read. When the file data is cached on the z/OS NFS client system and another client user on this system attempts to access the same file data, the z/OS NFS client checks the returned permission codes to validate access. Since the z/OS NFS server has passed a code which allows anyone access to the file, all users on the client system can

access the cached data without further restrictions. If data caching is turned off, no client caching takes place and each user must pass the server security check.

Based on the installation time out values, the file data cached by the client is flushed and further attempts to access the file data again requires passing server authorization.

The installation **DataCaching** parameter can be set and it can be overridden for each mount point so that different mount points can be handled as required for the files under that mount point.

If the potential security exposure can not be tolerated for sensitive file data, the **DataCaching** should be used so that no file data is cached by the z/OS NFS client.

## Mount processing parameters and installation parameters

Table 20 shows mount processing parameters.

*Table 20. Mount processing parameters*

acdirmax(n)	proto(tcp udp)
acdirmin(n)	ReadAhead(n)
acregmax(n)	retrans(n)
acregmin(n)	retry(n)
AttrCaching(Y N)	rsize(n)
cln_ccsid(n)	srv_ccsid(n)
DataCaching(Y N)	timeo(n)
DelayWrite(n)	vers(2 3)
delim (binary nl cr lf crlf lfcr)	wsiz(n)
DynamicSizeAdj(Y N)	xlat(Y N)
hard soft	

Table 21 shows installation parameters.

*Table 21. Installation parameters*

AttrCaching(Y N)	delim (binary nl cr lf crlf lfcr)
Biod(n)	DynamicSizeAdj(Y N)
Bufhigh(n)	ReadAhead(n)
cln_ccsid(n)	srv_ccsid(n)
DataCaching(Y N)	xlat(Y N)
DelayWrite(n)	

The following conditions may cause the NFS client to fail its initialization:

- The NFS client is not started in a standalone colony address space.
- The NFS client is already started; multiple instances of the NFS client on a single multiple virtual system (MVS) is not supported.
- Invalid parameter is specified in the installation parameters.
- If Unicode exists, then Unicode is used. If Unicode does not exist and Character Data Representation Architecture (CDRA) exists, then CDRA is used. If both Unicode and CDRA do not exist, then initialization fails.

A WTO message is issued to the operator console if the NFS client fails to initialize.

---

## Chapter 10. Attributes for the z/OS NFS server

This chapter contains information about the attributes that are used to manipulate files in the z/OS NFS server.

Table 22 contains directive information about this chapter's contents:

Table 22. Attributes - z/OS NFS server

Section	Modification	Description	Page
Data set creation attributes	Data set creation attributes can be modified by the client	Data set creation attributes provide information about an MVS file to the z/OS NFS server, such as the type of file, or how the file is allocated (for example, blocks, cylinders, or tracks)	76
Processing attributes	Processing attributes can be modified by the client	Processing attributes provide information to the z/OS NFS server about how to handle the file, such as how long the files remain open, or whether the files are processed in text or binary format	78
Site attributes	Site attributes can only be modified by the system administrator	Site attributes control the z/OS NFS server resources	86

---

### Attributes used for z/OS UNIX System Services file access

These attributes are specific to z/OS UNIX file access:

- `hfs(prefix)`
- `sync` and `async`
- `extlink`

These attributes are relevant to accessing z/OS UNIX files as well as conventional MVS data sets:

- `restimeout` - Resource timeout
- `logout` - User log time out
- `security` - Security checking
- `text` - ASCII to EBCDIC data conversion and vice versa
- `binary` - No ASCII and EBCDIC
- `xlat` - Customized translation table

---

### Using multipliers

Instead of entering the entire numeric values for the attributes, you can use the multipliers K (1024), M (1024 × 1024), or G (1024 × 1024 × 1024). For example, entering 10M is the same as entering 10,485,760.

---

### Specifying attributes multiple times

Specifying an attribute several times on a line does not cause an error. The line is read from left to right, and the last of any duplicate attribute is used. For example:

```
$ vi "file,recfm(vb),recfm(fb)"
```

This results in a file created with a fixed-blocked format.

---

## Data set creation attributes

The data set creation attributes are used to define the structure of MVS data sets when creating a file. These attributes correspond to the data control block (DCB) or the job control language (JCL) parameters used to define an MVS data set when it is created. See *z/OS MVS JCL Reference* for more information about data set creation attributes.

The data set creation attributes are described in Table 23. The data set creation attributes do not apply for HFS data sets. Defaults are underlined **in this format**.

You can override these attributes by using the **mount** command or file creation command. For PDS and PDSE, members have the same attributes as the data set attributes, so the file creation attributes specified for members are ignored.

Table 23. Data set creation attributes

Data Set Creation Attribute	Description
<b>blks</b>	Specifies that disk space (see the <b>space</b> attribute in this table) is allocated by blocks, except for VSAM data sets.
<b>cyls</b>	Specifies that disk space is allocated by cylinders.
<b>recs</b>	Specifies that disk space is allocated by records for VSAM data sets. <b>blks</b> and <b>recs</b> are identical for VSAM data sets.
<b>trks</b>	Specifies that disk space is allocated by tracks.
<b>blksize(0   <i>quan</i>)</b>	Specifies the maximum length, in bytes, of a physical block on disk. <i>quan</i> is a number from <u>0</u> (the default) to 32,760. If <b>blksize(0)</b> is specified, the system determines an optimal block size to use.
<b>dataclas(class_name)</b>	Specifies the data class associated with the file creation. The <i>class_name</i> must be defined to DFSMS/MVS® before it can be used by the client. The system-managed storage automatic class selection routine must also assign a storage class to the file being created. For more information about data classes, see <i>z/OS DFSMSdfp Storage Administration Reference</i> .
<b>dir(27   <i>quan</i>)</b>	Specifies the number of 256-byte records needed in the directory of a PDS. Use it with the <b>mkdir</b> command when you are creating a PDS. <i>quan</i> is a number from 1 to 16,777,215 (the default is <u>27</u> ). The maximum number of PDS members is 14,562.
<b>dsntype(library   <i>pds</i>)</b>	Specifies whether a PDSE or a PDS is to be created when the <b>mkdir</b> client command is used. <b>library</b> is for PDSE. <b>pds</b> is for PDS. You cannot create a PDS (or PDSE) within another PDS (or PDSE). If you need help deciding whether to create a PDS or a PDSE, see <i>z/OS DFSMS: Using Data Sets</i> .
<b>dsorg(org)</b>	Specifies the organization of a data set. <i>org</i> can be a physical sequential ( <b>ps</b> ) data set, direct access ( <b>DA</b> ) data set, VSAM KSDS ( <b>indexed</b> ), VSAM RRDS ( <b>numbered</b> ) or VSAM ESDS ( <b>nonindexed</b> ). This attribute is ignored for directory-oriented client commands. If you are using VSAM data sets in binary mode with an AIX client, then <b>nonindexed</b> is recommended.

Table 23. Data set creation attributes (continued)

Data Set Creation Attribute	Description
<b>keys</b> ( <i>len</i> , <i>off</i> )	Specifies the length and offset of the keys for VSAM KSDS data sets. Keys can only be specified when using <b>dsorg(indexed)</b> . <i>len</i> and <i>off</i> are specified in bytes. <i>len</i> is between 1 and 255 (the default is <b>64</b> ). <i>off</i> is between 0 and 32,760 (the default is <b>0</b> ). When you create a VSAM KSDS data set, the records you are loading into it must be keyed-sequenced or the write fails. Each write of the data set is treated like a first load, and requires that the records being loaded are in ascending key sequence.
<b>lrecl</b> ( <b>8196</b>   <i>quan</i> )	Specifies: <ol style="list-style-type: none"> <li>1. The length, in bytes, for fixed-length records.</li> <li>2. The maximum length, in bytes, for variable-length records. If the <b>blksize</b> attribute is specified, the value must be at least 4 bytes less than the <b>blksize</b> quantity.</li> </ol> <p><i>quan</i> is a number from 1 to 32,760 (the default is <b>8196</b>).</p>
<b>mgmtclas</b> ( <i>mgmt_class_name</i> )	Specifies the management class associated with the file creation. The <i>mgmt_class_name</i> must be defined to DFSMS/MVS before it can be used by the client. The system-managed storage automatic class selection (ACS) routine must also assign a storage class to the file being created. For more information on management classes, see <i>z/OS DFSMSdfp Storage Administration Reference</i> .
<b>recfm</b> ( <i>cccc</i> )	Specifies the format and characteristics of the records in the data set. <i>cccc</i> can be 1 to 4 characters, in one of the following combinations:  f   fb   fs   fbs u v   <b>vb</b>   vs   vbs  Valid record format characters: <b>b</b> Blocked <b>f</b> Fixed-length records <b>s</b> Spanned for variable records, standard format for fixed records <b>u</b> Undefined-length records <b>v</b> Variable-length records  In <b>recfm</b> , codes <b>v</b> , <b>f</b> and <b>u</b> are mutually exclusive. The <b>s</b> code is not allowed for a PDS or PDSE.
<b>recordsize</b> ( <i>avg</i> , <i>max</i> )	The average and maximum record size for VSAM data sets. <i>avg</i> and <i>max</i> are specified in bytes. They can each range from 1 to 32,760 (the defaults are <b>512</b> and <b>4096</b> , respectively). <b>These values must be equal for VSAM RRDS.</b>
<b>rlse</b>	Specifies that unused space should be released from the data set the first time a new data set is closed. For slow clients with long pauses between writes, the <b>rlse</b> attribute causes space to be released from the primary extent prematurely. Further writes cause secondary space to be allocated.
<b>norlse</b>	Specifies that unused space should not be released from the data set.
<b>shareoptions</b> ( <i>xreg</i> , <i>xsys</i> )	Specifies the cross-region and cross-system share options for a VSAM data set. <i>xreg</i> is a number from 1 to 4; <i>xsys</i> is either 3 or 4. The defaults are <b>1</b> and <b>3</b> , respectively.

Table 23. Data set creation attributes (continued)

Data Set Creation Attribute	Description
	This applies to VSAM data sets only. For spanned records of non-VSAM data sets, see the entry for <b>recfm</b> in this table.
<b>spanned</b>	Specifies that VSAM KSDS or ESDS data sets can contain records that span control intervals (spanned records).
<b>nonspanned</b>	Specifies that data sets do not have spanned records.
<b>space(prim[,aux])</b>	Specifies the amount of primary and auxiliary space allocated for a new data set on a direct access volume. <i>prim</i> is the number (from 0 to 16,777,215) of primary tracks, cylinders, or data blocks in the data set. <i>aux</i> (optional) is the number (from 0 to 16,777,215) of additional tracks, cylinders, or blocks allocated if more space is needed. If this attribute is not specified, the default is used. The defaults are <u>100</u> and <u>10</u> , respectively.
<b>storclas(class_name)</b>	Specifies the storage class associated with the file creation. The <i>class_name</i> must be defined to the DFSMS/MVS before it can be used by the client. For more information about storage classes, see z/OS DFSMSdfp Storage Administration Reference.
<b>unit(unit_name)</b>	Specifies the unit on which to create a data set. <i>unit_name</i> is a generic or symbolic name of a group of DASD devices. The <i>unit_name</i> must be specified as 3390 for extended format data sets. <b>Note:</b> You cannot create or access tape data sets on an MVS host using the z/OS NFS server. You cannot create extended format data sets with the z/OS NFS server, except using ACS routines.
<b>vol(volser)</b>	Specifies the name of the DASD volume to be used to store the created data set. <b>vol</b> is the keyword and <i>volser</i> represents the volume name. If a data set is to be system-managed, as is determined by the DFSMS/MVS Automatic Class Selection (ACS) routines, you can omit this attribute.

## Processing attributes

Processing attributes are used to control how files are accessed by the client. The processing attributes are described in Table 24. Defaults are underlined **in this format**. You can override the default processing attributes.

Table 24. Processing attributes

Processing Attribute	Description
<b>attrtimeout(n)</b>	The time (in seconds) that the data set remains allocated after a <b>lookup</b> or <b>getattr</b> server operation. The default is <u>120</u> . <i>n</i> can range from 1 to 32,767 (9 hours, 6 minutes, and 7 seconds). <b>Note:</b> <b>lookup</b> is an NFS protocol that searches for a file in the current directory. If it finds the file, <b>lookup</b> returns information on the file's attributes and a file handle pointing to the file.  The <b>attrtimeout</b> value is normally greater than the <b>readtimeout</b> or <b>writetimeout</b> values.
<b>noattrtimeout</b>	The data set is not deallocated after a <b>lookup</b> or <b>getattr</b> operation.  For more information, see "Timeout attributes" on page 82, following this table.

Table 24. Processing attributes (continued)

Processing Attribute	Description
<b><u>binary</u></b>	Indicates that the data set is processed between the client and server using binary format and no data conversion occurs between ASCII and EBCDIC formats.
<b>text</b>	<p>Converts the contents in the data set between EBCDIC and ASCII formats. Use this format to share text data between clients and MVS applications.</p> <p>In text mode, the following attributes apply only to conventional MVS data sets:</p> <ul style="list-style-type: none"> <li>• <b><u>blankstrip</u></b> and <b>nobblankstrip</b>. See the entry for <b><u>blankstrip</u></b> in this table.</li> <li>• End-of-line specifiers (<b><u>If</u></b>, <b>cr</b>, <b>lfcr</b>, <b>crlf</b>, or <b>noeol</b>) are used to indicate the MVS logical record boundary. See the entry for <b><u>If</u></b> in this table. See the <b>xlat</b> attribute in this table for customized EBCDIC-ASCII tables.</li> </ul>
<b><u>blankstrip</u></b>	<p>With text mode, strips trailing blanks at the end of each record of a fixed-length text file when the file is read. Pads the end of each file or record with blanks when a text file is written.</p>
<b>nobblankstrip</b>	<p>Does not strip trailing blanks at the end of fixed-length records when a fixed-length text file is read. Does not pad records when writing a text file. The file must be of the correct size or an I/O error is reported to the client.</p> <p>For information on the <b>text</b> attribute, see the entry for <b><u>binary</u></b> in this table.</p> <p>This attribute does not apply to HFS data sets.</p>
	<p>With text mode, use one of the following end-of-line specifiers:</p>
<b><u>If</u></b>	Line Feed is the end-of-line terminator (standard AIX or UNIX).
<b>cr</b>	Carriage Return is the end-of-line terminator.
<b>lfcr</b>	Line Feed followed by Carriage Return is the end-of-line terminator.
<b>crlf</b>	Carriage Return followed by Line Feed is the end-of-line terminator (standard DOS).
<b>noeol</b>	No end-of-line terminator.
	<p>For information on the <b>text</b> attribute, see the entry for <b><u>binary</u></b> in this table.</p> <p>This attribute does not apply to HFS data sets.</p>
<b>cln_ccsid(n)</b>	<p>Specifies the Coded Character Set Identifier (CCSID) for the remote mounted file system (NFS client) when text is being translated.</p> <p>The default is <b><u>819</u></b> (ISO 8859-1 ASCII).</p> <p>This attribute applies only to hierarchical file system (HFS) files.</p>
<b>executebiton</b>	<p>Turns on the execute bits in user, group, and other (as reported with the <b>ls</b> (list) AIX or UNIX command) for a mount point's files. Use when storing executable or shell scripts on the MVS system. This option can only be overridden on a mount point basis — not at a command level. This attribute does not apply to HFS files and can only be used with the mount command.</p>
<b><u>executebitoff</u></b>	<p>Turns off the execute bits in user, group, and other for the mount point's files. This value is normally used in the site file.</p>

Table 24. Processing attributes (continued)

Processing Attribute	Description
<b>extlink</b>	Specifies the use of the external link command to create, process, and delete a symbolic link to an MVS data set. Is used with the <b>ln -s</b> command to create a symbolic link to an MVS data set. Is used with the <b>ls -l</b> command to display the attributes and contents of the symbolic link. Is used with the <b>rm</b> command to delete the symbolic link.  This <b>extlink</b> attribute only applies to HFS file objects.
<b>fastfilesize</b>	For Direct Access data sets, PDSs, and non-system-managed data sets, this specifies to get the file size from SPF statistics, if they exist. For more information, see "Using fastfilesize to avoid read-for-xize" on page 144. This attribute also applies to PDSEs, but does not apply to HFS files.
<b>nofastfilesize</b>	For Direct Access data sets, PDSs, and non-system-managed data sets, this specifies to read the entire file or member to get the file size. Using this attribute might cause a noticeable delay when first accessing very large data sets. For more information, see "Using fastfilesize to avoid read-for-xize" on page 144. This attribute also currently applies to PDSEs.
<b>fileextmap</b>	Turns on file extension mapping. This option can be specified at the file command level for the client platforms that support passing of attributes. The default is <b>nofileextmap</b> .
<b>nofileextmap</b>	Turns file extension mapping off.
<b>mapleaddot</b>	Turns on mapping of a single leading "." from a client file name to a legal leading "\$" on MVS. This option would normally be enabled for access by AIX and UNIX clients. This attribute does not apply to HFS data sets.
<b>nomapleaddot</b>	Turns off mapping of a single leading "." from a client to a leading "\$" on MVS.
<b>maplower</b>	Turns on mapping of lower case file names to upper case when accessing files on MVS, and back when sending to the network. This option would normally be enabled for access by AIX or UNIX clients. This option only affects file names (high-level qualifiers and user catalog aliases). This attribute does not apply to HFS data sets.
<b>nomaplower</b>	Turns off mapping of lower case file names to upper case and back when using files on MVS. Exports data set entries are not translated to upper case when this option is specified in the default attributes. All <b>mount</b> requests are case sensitive.
<b>mapped</b>	This attribute should be specified at the mount or site level when a mixed set of data types is to be processed under a single mount point. The determination of whether the data is to be processed as text or binary depends on the rules that are established in the specified sidefile. If a file extension is not mapped to text or as binary using the sidefiles, then the data will be processed according to what has been specified as binary or text at the mount level, and finally, at the site level. If binary or text is specified at the file command level, the specification overrides the mapped specification.
<b>rdrrverf</b>	Do cookie verifier checking for NFS version 3 <i>readdir</i> and <i>readdirplus</i> requests.
<b>nordrverf</b>	Do not do cookie verifier checking for NFS version 3 <i>readdir</i> and <i>readdirplus</i> requests.



Table 24. Processing attributes (continued)

Processing Attribute	Description
<b><u>readtimeout(<i>n</i>)</u></b>	<p>The amount of time in seconds before a data set is released after a read operation. <i>n</i> can range from 1 to 32,767 (9 hours, 6 minutes, and 7 seconds). The default is <b>90</b>. The server closes the file when the file times out.</p> <p>This attribute does not apply to HFS.</p>
<b><u>noreadtimeout</u></b>	<p>The data set is not deallocated after a read operation.</p> <p>For more information, see “Timeout attributes” on page 82, following this table.</p>
	<p>The z/OS NFS server uses DFSMSHsm™ to recall or delete migrated files. The action that the server takes against the migrated files depends on which of the <b>retrieve</b> or <b>noretrieve</b> attributes is active. These attributes do not apply to HFS data sets.</p>
<b><u>retrieve</u></b>	<p>When the <b>retrieve</b> attribute is active, the server will recall the migrated file if necessary, upon an NFS_LOOKUP request for the file, depending on the files status. The server may be able to obtain the migrated files attributes without recall (see “Retrieve attributes” on page 83 for additional information); if not the recall operation is started by the server. The server waits for the recall operation to complete if the file resides on DASD; if the file does not reside on DASD, the server does not wait for the recall operation to complete and returns a “device not available” message. You can try accessing the file later when the recall has completed.</p>
<b><u>retrieve(wait)</u></b>	<p>When the <b>retrieve(wait)</b> attribute is active, the server waits for the recall to finish.</p>
<b><u>retrieve(nowait)</u></b>	<p>When the <b>retrieve(nowait)</b> attribute is active, the server does not wait for the recall to finish, and immediately returns a “device not available” message. You can try accessing the file later when the recall has completed.</p>
<b><u>noretrieve</u></b>	<p>When the <b>noretrieve</b> attribute is active, the server does not recall the file, and an return “device not available” upon an NFS_LOOKUP, an NFS_READ, or an NFS_CREATE request for a file.</p> <p>For more information, see “Retrieve attributes” on page 83, following this table.</p>
<b><u>setownerroot</u></b>	<p>Sets the user ID in a file’s attributes to root for a superuser. This attribute can only be used with mount command and does not apply to HFS.</p>
<b><u>setownernobody</u></b>	<p>Sets the user ID in a file’s attributes to nobody, for a superuser.</p>
<b><u>sidefile(<i>dsname</i>)</u></b>	<p>Specifies the name of the data set that contains the rules for file extension mapping purposes. If a sidefile name is specified in the attributes data set, then it is the default sidefile for the NFS server. A user can also specify an additional sidefile name during a mount operation to be used along with the default. The mapping rules will first be searched in the sidefile specified during the <b>mount</b> command and then the default sidefile is searched. To allow file extension mapping, a sidefile name must be specified either as a default or in the mount command. <i>dsname</i> is a fully-qualified MVS data set name without quotation marks. <i>sidefile</i> is only specified at the mount level. See GFSAPMAP in NFSSAMP for sample sidefile and syntax rules. This attribute does not apply to HFS.</p>

Table 24. Processing attributes (continued)

Processing Attribute	Description
<b>srv_ccsid(n)</b>	<p>Specifies the Coded Character Set Identifier (CCSID) for the local mounted file system (NFS server) when a new file is being created.</p> <p>This attribute has no effect on the translation of existing files' data.</p> <p>The default (if specified) is <b>1047</b> (LATIN OPEN SYSTEM EBCDIC).</p> <p>If this attribute is not specified, new hierarchical file system (HFS) files will continue to be created as untagged.</p> <p>This attribute applies only to HFS files.</p>
<b>sync</b>	Specifies that data transmitted with the <b>write</b> request should be committed to nonvolatile media (for example, DASD) by the server when received.
<b>async</b>	<p>The user can alternatively specify the <b>async</b> processing attribute to get improved performance.</p> <p>The <b>sync async</b> attribute only applies to HFS file objects and the NFS version 2 protocol.</p>
<b>writetimeout(n)</b>	<p>Specifies the amount of time, in seconds, before a data set is released after a write operation. <i>n</i> can range from 1 to 32,767 (9 hours, 6 minutes, and 7 seconds). The default is <b>30</b>. The server closes the file when the file times out. All cached buffers are forced to disk. Normally <b>writetimeout</b> values are kept short because WRITE operations result in exclusive locking. However, for slow client machines with long pauses between writes, you should increase the <b>writetimeout</b> value.</p> <p>This attribute does not apply to HFS.</p>
<b>nowritetimeout</b>	<p>Specifies that the data set is not deallocated after a write operation.</p> <p>For more information, see "Timeout attributes", following this table.</p>
<b>xlat(member_name)</b>	<p>This attribute can be used to override the installation default translation table during file processing. <i>member_name</i> is the member name of the PDS or PDSE that contains the customized translation table. The system administrator defines this member name in the attribute data set, and PDS or PDSE in the startup procedure. This attribute is ignored if specified on the command line. See Customizing the Translation Table in the z/OS NFS Customization and Operation Manual for more information.</p> <p>If a customized translation table is not specified in the attribute file or in the mount command, <b>xlat()</b> is displayed by the <b>showattr</b> client enabling command.</p>

## Timeout attributes

The values of the following attributes depend on the settings of the associated site attributes:

- The attributes **attrtimeout**, **readtimeout**, and **writetimeout** must be within the ranges specified by the **maxtimeout** and **mintimeout** site attributes.
- The attributes **noattrtimeout**, **noreadtimeout** and **nowritetimeout** are valid only when **nomaxtimeout** is specified in the site attributes.

There are three processing attributes which control when files are timed out: **attrtimeout**, **readtimeout**, and **writetimeout**. The server determines that of these timeouts are in effect based on the last file operation. Thus when an existing file is appended, the file cannot be accessed before it times out in the time specified for **writetimeout** and is released by the server, because write operations result in

exclusive locking. Similarly, if a file is read, it is not released before it times out in the time specified for **readtimeout** seconds.

The new **readdirtimeout** site attribute controls the internal **readdir** cache used by directory lookups of MVS conventional data sets to be timed out or discarded based on a customizable value. The default is 30 seconds.

## Retrieve attributes

The server deletes the migrated file upon an NFS\_REMOVE request for a file, regardless of whether the **retrieve** or the **noretrieve** attribute is active. Typically, an NFS\_REMOVE request is preceded by an NFS\_LOOKUP request. If the data set was migrated with DFSMS/MVS 1.2 or below, retrieve attribute causes a recall because NFS\_LOOKUP processing needs to open the data set and read for size. If the data set was migrated under DFSMS/MVS 1.3 and DFSMSHsm 1.3, and is SMS managed, its attributes were saved on DASD; therefore it is not always necessary to recall the data set to read for size and the data set may be deleted without recall. If the **noretrieve** attribute is active, the NFS\_LOOKUP can return a “device not available” message. If the client code decides to ignore the error and go with the NFS\_REMOVE, the migrated file is then deleted.

The UNIX command **ls mvshost** does not issue requests for individual files under the mvshost directory. Migrated files under the mvshost directory are displayed, but are not recalled. However, the UNIX command **ls -l mvshost** issues NFS\_LOOKUP requests for individual files under the mvshost directory.

## Mapped keyword processing attribute

Table 25 contains *mapped* and existing keyword information:

Table 25. The mapped keyword and existing keywords

SFMAX	SIDFILE(NAME)	MAPPED	ACTION
=0	X	Don't care	Data processed using existing rules for binary/text
	I		Server won't come up
	M		MOUNT will fail

Table 25. The mapped keyword and existing keywords (continued)

SFMAX	SIDFILE(NAME)	MAPPED	ACTION
1–2000	I + M	SET	File extension used in the MOUNT-specified sidefile and then the site-specified sidefile. If an extension is not found, the existing rules for binary/text will be used.
		NOT SET	Data processed using existing rules for binary/text
	I	SET	File extension used in the site-specified sidefile. If an extension is not found, the existing rules for binary/text will be used.
		NOT SET	Data processed using existing rules for binary/text
	X	Don't care	Data processed using existing rules for binary/text
	M	SET	File extension used in the mount-specified sidefile. If an extension is not found, the existing rules for binary/text will be used
		NOT SET	Data processed using existing rules for binary/text

Legend:

I = sidefile specified in installation table

M = sidefile specified in mount command

X = no sidefile specified

## Native ASCII processing attributes

The **cln\_ccsid(n)** and **srv\_ccsid(n)** attributes can be specified either as installation defaults or at mount time for more granularity between different mount points. Unless **srv\_ccsid** is specified either as an installation default or at mount time, newly created files will not have any file tag set (i.e., file tag is all zeros). These two attributes affect translation only when text processing is involved and only when an existing file has a non-zero or a non-binary file tag.

Special attention must be paid to the different server attributes specified. See Table 26 on page 85 and Table 27 on page 85.

### Considerations for Native ASCII environment support

For applications running on z/OS V1R2 (and higher), a Native ASCII environment is provided for hierarchical file system (HFS) file processing.

In this environment, applications can operate on files in either EBCDIC or ASCII format as well as other data formats defined with a Coded Character Set Identifier (CCSID) without translation, provided the data is already defined and stored in the data format wanted.

For the z/OS NFS server to operate properly on HFS files in this environment, consider the following important factors:

- Unicode Conversion Services must be installed and set up on the system to let the NFS server use it for text translation.

- Two new processing attributes, **cln\_ccsid** and **srv\_ccsid**, are available for the NFS server for translation purposes as well as for the creation of new files. The **srv\_ccsid** attribute determines the CCSID of newly created HFS files. If **srv\_ccsid** is not specified as an installation default or at mount time, then new files continue to be created as untagged, or with a tag of 0x0000 and the old translation method of using translation tables specified by the **xlat** keyword applies
- Processing (READ/WRITE) of tagged files depends on the different server options specified. See details in “Native ASCII processing attributes” on page 84.

Table 26 and Table 27 contain considerations for Native ASCII environment support.

**NFS clients with non-390 based NFS servers:** Table 26 contains NFS client options.

Table 26. z/OS NFS clients with non-390 based NFS servers

Client Options Specified	Mount Option	Read	Write
xlat(Y), cln_ccsid,srv_ccsid	No TAG specified	Same as today - NFS client does translation	Same as today - NFS client does translation
xlat(N)	TAG(TEXT,CCSID)	Logical file system does translation	Logical file system does translation
xlat(Y)	TAG(TEXT,CCSID)	Mount will fail	Mount will fail

**Notes:**

1. The logical file system will do translation when the **mount TAG** option is specified. It will do translation based on the process tag (calling application) and file tag (if the file tag is not zeros or untagged). Otherwise, the system will do translation based on process tag and mount TAG the CCSID information.
2. It is assumed that the user doing the **mount** knows the files being accessed from the remote non-390 file systems. So the CCSID needs to be set accordingly. Data written to the server will be stored in a specific CCSID format and to read it back correctly, the correct CCSID needs to be specified (for example, without it being translated with the wrong CCSID).

See *z/OS Network File System User's Guide* for more information about client mount options.

**NFS servers with non-390 based NFS clients:** Table 27 contains NFS server options. See “Native ASCII processing attributes” on page 84 for more information.

Table 27. NFS servers with non-390 based NFS clients

Server Options Specified	File Tag	Read	Write	Create
text <xlat(table_name)> <sup>5</sup>	Untagged or Tag=0x0000	Translation using the current xlat tables	Translation using the current xlat tables	New file created with CCSID of 0x0000
text	Yes	Translation based on file tag and default cln_ccsid of 8859 <sup>3</sup>	Translation based on file tag and default cln_ccsid of 8859 <sup>3</sup>	New file created with CCSID of 0x0000
text xlat(table_name)	Yes	Fail operation	Fail operation	New file created with CCSID of 0x0000
text,cln_ccsid,<srv_ccsid> <xlat(table_name)> <sup>5</sup>	Untagged or Tag=0x0000	Translation using the current xlat tables	Translation using the current xlat tables	New file created with srv_ccsid tag <sup>2</sup>
text,cln_ccsid,<srv_ccsid> <sup>4</sup>	Yes	Translation based on file tag and cln_ccsid <sup>3</sup>	Translation based on file tag and cln_ccsid <sup>1,3</sup>	New file created with srv_ccsid tag <sup>2</sup>

Table 27. NFS servers with non-390 based NFS clients (continued)

Server Options Specified	File Tag	Read	Write	Create
binary	Untagged or Tag=0x0000	No translation	No translation	New file created with binary tag
binary	Yes	No translation	Fail operation unless file tag is binary or TXTFLAG=OFF in file tag	New file created with binary tag

**Notes:**

1. Writing to a file that has a tag that is different from the **srv\_ccsid** (regardless of whether the file is empty or not) will result in the file tag overriding the specified **srv\_ccsid** when text is specified.
2. If **srv\_ccsid** is specified (as an installation default or at mount), then the file will be created with the **srv\_ccsid** tag. Otherwise, an untagged file is created.
3. If TXTFLAG=OFF (that is, not pure text) with non-zero CCSID and text is specified, it is assumed that the user knows which part of the file is text and translation proceeds as requested by client. In order to access a file that has mixed data without the data being translated during read or write operation, the binary option must be used.
4. **xlat** (if specified) is ignored when the file being accessed is tagged.
5. **xlat** is optional. For untagged files, translation is done using default **xlat** tables or customized **xlat** tables (if specified).
6. There is no facility in the NFS server to change an existing file tag. This must be done outside of the NFS server.
7. Specifying the binary option overrides any **cln\_ccsid** and **srv\_ccsid** is specified.
8. All files created that are by the server when text and **srv\_ccsid** are specified will also have the TXTFLAG set to ON.

**z/OS NFS client with z/OS NFS server:** Both the client and server operate as described in “NFS clients with non-390 based NFS servers” on page 85 and “NFS servers with non-390 based NFS clients” on page 85.

In order to avoid double translation, the mount to the server must specify the correct **cln\_ccsid** (server option) and the client **tso mount** command should not have the **tag** option. The client mount option **xlat(N)** should be specified so that only the server will do translation (if needed) and return the data in the correct CCSID.

```
mount filesystem(NFS001) type(nfs) mountpoint('/u/nfsdir')
      parm('mvsnfs:"/hfs/u/user,text,cln_ccsid(2000)",xlat(N)')
      vi /u/nfsdir/file1

** Translation will be done based on file1's file tag and cln_ccsid of
   2000 by server only.
```

In all other cases, double translation may occur as the server will do translation based on its file tag and **cln\_ccsid** settings and logical file system will do translation based on the process tag and the CCSID in the **mount TAG** option. Caution must be used as double translation may result in the data becoming garbage.

## Site attributes

The site attributes are used to control z/OS NFS server resources. These attributes are described in Table 28 on page 87. Some initial settings are shown, but the system administrator might have changed these settings, so use the **showattr** command to show the actual settings being used. The site attributes cannot be modified by client users.

Table 28. Site attributes

Site Attribute	Description
<b>bufhigh(<i>n</i>)</b>	<p>Specifies the maximum size (in bytes) of allocated buffers before buffer reclamation (see the <b>percentsteal</b> attribute in this table) is initiated. <i>n</i> is an integer from 1 MB to 2047 MB (the default is <b>32</b> MB). If the combined total specified in the <b>bufhigh</b> and <b>logicalcache</b> attributes is greater than the available storage in the extended private area (implied by the REGION parameter in your procedure) at startup, the server shuts down immediately. A higher number means more caching and potentially better read performance.</p>
<b>cachewindow(<i>n</i>)</b>	<p>Specifies the window size used in logical I/O to buffer client block writes received out of order. <i>n</i> is a number from 1 to 256 (the default is <b>112</b>). This attribute does not apply to HFS data sets. The suggested value is some small multiple of the number of BIODs running on a client. The general rule in setting the <i>n</i> value of cachewindow(<i>n</i>) is <math>n = ((\text{num of BIOD} + 1) * (\text{client\_max\_IO\_buffer\_size}/\text{transfer\_size}))</math> where,</p> <ul style="list-style-type: none"> <li>• BIOD is the number of blocked I/O daemons set by the client workstation. This value is usually set to defaults at the installation of the operating system or by your system administrator.</li> <li>• client_max_IO_buffer_size is the amount of I/O data requested by the client (for example, client writes 8192 bytes of data to the remote file system). This value is determined by your application programs.</li> <li>• transfer_size is the actual size of data being sent across the network (for example, the 8192 bytes of data can be broken down to 16 smaller packets of 512 bytes (<math>16 \times 512 = 8192</math>)). This value is determined dynamically by your client workstation.</li> </ul>
<b>checklist</b>	<p>When specified, the server bypasses SAF checking (even when SAF or SAFEXP is specified) for the list of files and directories underneath mount points which either matches a mount point entry or is a child of a mount point entry in the CHKLIST DD data set. CHECKLIST is only valid if SAF checking is the security option for the particular data access; otherwise, it is ignored even if it is specified. See GFSAPCHK in NFSSAMP library for sample CHKLIST data set.</p>
<b><u>nochecklist</u></b>	<p>When specified, the server operates as before and ignores the information that is specified in the CHKLIST DD data set.</p>
<b>fn_delimiter</b>	<p>Specifies a character 'c' to be used as a delimiter between the file name and the attributes that follow it. This capability allows those sites that have UNIX data sets containing commas to copy and store their data on the NFS server. The following example specifies the default delimiter as a semicolon:</p> <pre>fn_delimiter(;)  So a user can process a file called 'comma,in-name' by entering:  vi "comma,in-name;text,lf"  A user can also include a default file name delimiter as a comma as follows:  fn_delimiter(,)</pre>
<b><u>fn_delimiter(,)</u></b>	<p>The default file name delimiter is a comma.</p>

Table 28. Site attributes (continued)

Site Attribute	Description
<b>hfs(prefix)</b>	<p>Specifies a new HFS file system <i>prefix</i> to be imbedded in the mount directory path name. The default value of the HFS file system <i>prefix</i> is <b>/hfs</b>. Mount requests received by the z/OS NFS server beginning with the HFS file system <i>prefix</i> value are identified as mount requests for z/OS UNIX. The HFS file system <i>prefix</i> value is not part of the path name.</p> <p><b>Notes:</b></p> <ol style="list-style-type: none"> <li>1. The HFS file system must be mounted locally by z/OS UNIX or the client mount fails.</li> <li>2. The <i>prefix</i> value can only be 7 characters or less including the beginning "/"</li> </ol>
<b>nohfs</b>	Disables Network File System z/OS UNIX operation.
<b>hfsbtimeout</b>	<p>Specifies how to control the timeout of the HFS vnode token used by the NFS server. The timeout value controls how long before vnode tokens saved in file blocks are released.</p> <p>The valid range is 1 to 32,767 seconds.</p> <p><i>n</i> can go as low as 1 second but to avoid the possibility of the client hanging (because of network dealys), <i>n</i> is not recommended to be lower than 5 seconds. <i>n</i> may need to be increased if the network is slow and the accessed directory has a lot of entries.</p> <p>The default <b>hfsbtimeout</b> is 60 seconds.</p>
<b>leadswitch</b>	Tells the server to return '/' as the first character in each export entry.
<b>noleadswitch</b>	<p>Tells the server not to return '/' as the first character in each export entry.</p> <p><b>Note:</b> The <b>leadswitch</b> attribute is ignored for HFS file objects.</p>
<b>logicalcache(n)</b>	<p>Specifies the maximum size (in bytes) of allocated buffers in the logical I/O processing for all the cache windows combined. <i>n</i> is an integer from 1 MB to 2047 MB (the default is <b>16 MB</b>). The recommended value for this attribute is 50% of the <b>bufhigh</b> attribute. This attribute does not apply to HFS data sets.</p>
<b>logout(n)</b>	<p>Specifies the time limit for inactivity in seconds for a given user on a client (the default is <b>1800</b>). When the limit is reached, the user is automatically logged out. The client user must enter the <b>mvslogin</b> command again to reestablish the client's MVS session. This value should normally be the same as the value defined for TSO logout at your site. <i>n</i> can range from 1 second to 20 megaseconds (approximately 243 days).</p>
<b>maxrdforszleft(n)</b>	<p>Defines the number of physical block buffers left after determining a file's size. This operation is done for later server read requests to the same file. The buffers left are subject to trimming during a "buffer steal" operation. <i>n</i> is an integer from 1 to 1024 (the default is <b>32</b>).</p>
<b>maxtimeout(n)</b>	<p>Specifies the maximum timeout allowed. This attribute and the <b>mintimeout</b> attribute define the range of values that client users can specify for <b>attrtimeout</b>, <b>readtimeout</b>, and <b>writetimeout</b>. <i>n</i> is the number of seconds from 1 to 32,767 (9 hours, 6 minutes, and 7 seconds). This attribute does not affect the <b>logout</b> attribute.</p>
<b>nomaxtimeout</b>	Allows client users to specify <b>noattrtimeout</b> , <b>noreadtimeout</b> , and <b>nowritetimeout</b> .



Table 28. Site attributes (continued)

Site Attribute	Description
<b>mintimeout(<i>n</i>)</b>	Specifies the minimum timeout. This attribute and <b>maxtimeout</b> define the range of values that can be specified for <b>attrtimeout</b> , <b>readtimeout</b> , and <b>writetimeout</b> . <i>n</i> is the number of seconds from 1 to 32,767 (the default is <u>1</u> ).
<b>nfstasks(<i>n,m,o</i>)</b>	<p>Specifies the number of server processes to initiate on startup. The valid value range for <i>n</i>, <i>m</i>, and <i>o</i> are as follows:</p> <p>If <b>nfstasks(<i>n,m</i>)</b> is specified, then the following is true: <i>n</i> is the number of subtasks that handle the asynchronous input/output (I/O) operations or short blocking operations (the maximum number of concurrent NFS server requests). <i>m</i> is the number of subtasks that handle the long blocking operations (the maximum number of concurrent NFS server recall and HFS requests). Increase this value if your server supports lots of active recall or HFS clients.</p> <p>The valid value range for both <i>n</i> and <i>m</i> is 1 to 24. The sum of <i>n</i> and <i>m</i> must be less than or equal to 25.</p> <p>If <b>nfstasks(<i>n,m,o</i>)</b> is specified, then the following is true: <i>n</i> is the number of subtasks that handle the asynchronous input/output (I/O) operations or short blocking operations (the maximum number of concurrent NFS server requests). <i>m</i> is the number of subtasks that handle HFS requests. Increase this value if your server supports lots of active HFS clients. <i>o</i> is the number of subtasks that handle the long blocking operations (the maximum number of concurrent NFS server recall requests). Increase this value if your server supports lots of active recall operations.</p> <ul style="list-style-type: none"> <li>• The valid value range for <i>n</i> is 1 to 24.</li> <li>• The valid value range for <i>m</i> is 1 to 100.</li> <li>• The valid value range for <i>o</i> is 1 to 24.</li> <li>• The sum of <i>n</i> and <i>o</i> must be less than or equal to 25.</li> </ul> <p>The default is <b>nfstasks(8,16,8)</b>.</p>
<b>pcnfsd</b>	Tells the Network File System to start the PCNFSD server.
<b>nopcnfsd</b>	Tells the Network File System not to start the PCNFSD server.
<b>percentsteal(<i>n</i>)</b>	Specifies the percent of the buffers reclaimed for use when the <b>bufhigh(<i>n</i>)</b> limit has been reached. A higher value means a reclaim operation is performed less often, but the cached data is significantly trimmed on each reclaim. This can result in poor read performance because readahead buffers might be stolen. Lower values result in more frequent reclaim operations, but the cached data normal water mark is higher, meaning possibly better performance by reading out of cached data. <i>n</i> is an integer from 1 to 99 (the default is <u>20</u> ).
<b>public(legacy_path,hfs_path)</b>	<p>This attribute specifies the legacy path (MVS conventional data) and HFS path that is associated with the public file handle for WebNFS access. The first path, if specified, is the legacy path. The second path is the HFS path and must start with the HFS prefix specified in the HFS() keyword. If the first path is not there, a comma must precede the second path. If the <i>public</i> keyword is specified, then one of the paths must be specified. The <i>public</i> keyword must be specified after the <i>HFS()</i> keyword in the site attribute table. A <b>lookup</b> request with the <i>public</i> file handle determines which of the two paths it is referring to by the pathname that it comes in with. An absolute pathname will tell the server which of the paths it is referring to by matching one of the paths specified. A <b>lookup</b> request with a relative pathname will be taken to be an HFS request if HFS is active (hfs_path has been provided); otherwise, it is treated as a legacy request. The default is <b><u>no public path</u></b></p>

Table 28. Site attributes (continued)

Site Attribute	Description
<b>readaheadmax(<i>n</i>)</b>	<p>This attribute defines the number of bytes to be read to fill internal buffers during read processing to enhance satisfying read requests directly from cache. This reduces the amount of synchronous physical I/O performed for Network File System read requests for sequential read file access. It also reduces context switching overhead on Network File System read requests by allowing more read requests to be satisfied directly from the main task. <i>n</i> is an integer from 1 KB to 128 KB (normally 2 to 4 times the common block size used for file access, which is recommended at 8 KB for AIX file activity). The default is <b>16,384</b>. Specifying zero (0) will deactivate readahead.</p>
<b>readdirtimeout</b>	<p>Specifies the amount of time, in seconds, before the internal <i>readdir</i> cache that is used for MVS conventional data set is timed out or discarded. Valid range is from 1 to 32,767 (9 hours, 6 minutes, and 7 seconds). <i>n</i> can go as low as 1 second, but to avoid the possibility of <i>client hanging</i> (due to network delays and staled cache), <i>n</i> is not recommended to be lower than 5 seconds. <i>n</i> may need to be increased if the network is slow and the accessed directory has a lot of entries.</p> <p>The default readdirtimeout is 30 seconds.</p>
<b>retimeout(<i>n,m</i>)</b>	<p>Specifies a retention period and a clock time for the removal of mount points and control blocks that have been inactive longer than the specified retention period.</p> <p><i>n</i> is the resource retention period for mounts and associated resources. If they have been inactive for more than <i>n</i> hours, they are removed. The valid range for <i>n</i> is 0 to 720 hours (30 days). The default is <b>48</b> hours. If <i>n</i> is set to 0, the z/OS NFS server does not remove any mount points or associated resources.</p> <p><i>m</i> is the time of day to do the cleanup for mounts and associated resources that have been inactive more than <i>n</i> hours. The time of day is specified as a 24 hour local time value. The valid range for <i>m</i> is 0 to 23. The default is <b>0</b> (that is, midnight).</p> <p>Because cleanup work slows down the server, set <i>m</i> so that cleanup work occurs when the server is lightly loaded. If a mount handle is removed by the cleanup activity, the user must do the <b>umount</b> and <b>mount</b> operations to access the mount point again. The resource cleanup is also done when the server is shutting down.</p>
<b>security(mvs,hfs,public)</b>	<p>Where MVS data sets (legacy data), HFS data, and public data (data that is accessed using the public file handle) are place holders for security options: NONE, SAF, SAFEXP, EXPORTS. The first parameter, <b>mvs</b>, specifies the security option for MVS conventional data. The second parameter, <b>hfs</b>, specifies the security option for HFS data. The third parameter, <b>public</b>, specifies the security option for data that is accessed with the public file handle. The <b>mvs</b> parameter is required and the <b>hfs</b> and <b>public</b> parameters are optional. When the optional parameters are not specified, they are assigned the same security option as the first parameter.</p> <p><b>Note:</b> The security options are the same as before (NONE, SAF, SAFEXP, EXPORTS); however, now a different security option can be picked for each of the three different data accesses.</p> <p>The security options are as follows:</p> <p><b>none</b> Neither SAF checking nor exports list checking. For HFS, checks UNIX permission bits; obtains the UID from the client RPC request.</p> <p><b>saf</b> SAF checking. No exports checking. For HFS, checks UNIX permission bits; obtains the UID from the z/OS UNIX segment using <b>mvslogin</b>.</p> <p><b>exports</b> Exports list checking. For HFS, checks UNIX permission bits for HFS data; obtains the UID from the client RPC request. No SAF checking.</p> <p><b>safexp</b> SAF checking and exports list checking. For HFS, checks UNIX permission bits; obtains the UID from the z/OS UNIX segment using <b>mvslogin</b>.</p> <p>The default is security(SAFEXP,SAFEXP, SAFEXP)</p>

Table 28. Site attributes (continued)

Site Attribute	Description
<b>sfxmax(n)</b>	Where <i>n</i> specifies the maximum size (in kilobytes) of allocated storage for all of the sidefiles. <i>n</i> is an integer from 0 to 2000. The default value is 0 and it signifies that no mapping is allowed on the NFS server. If sfxmax=0, specifying <b>sidefile</b> keyword in the attributes data set will cause the server to shut down and the specifying the sidefile in any subsequent mount commands causes the mount to fail as mapping is not allowed on the NFS server. If the amount of storage specified cannot be obtained during server initialization then the server will shut down immediately.
<b>smf(none user file userfile)</b>	Specifies the SMF records to be recorded: <ul style="list-style-type: none"> <li><b>none</b> No SMF data collection</li> <li><b>user</b> Collection of user session SMF data</li> <li><b>file</b> Collection of file usage SMF data</li> <li><b>userfile</b> Collection of file usage and user session SMF data</li> </ul>



---

## Appendix A. Network File System messages

This appendix lists messages from the NFS server, the NFS client, and the client operating system.

Table 29 contains directive information about this chapter's contents:

*Table 29. Messages - client operating system, NFS server, and NFS client*

Section	Page
Server messages	93
Client messages	118
Messages from the client platform (AIX)	135

---

### Server messages

This is a listing of the messages generated by the NFS server. For each message, explanations and recommended actions are given where applicable. Data is substituted for any part of a message shown here in *italics*.

Messages that appear in the NFS server log data set in the same format as this example: **19:25:14 GFSA348I (I) GFSAMAIN ANMAI 02 NFS\_INIT: OS/390 VERSION 1 NETWORK FILE SYSTEM SERVER (HDZ11TS) STARTED.**

Table 30 shows the NFS server log data set message format:

*Table 30. NFS server log data set message format*

<b>19:25:14</b>	The time stamp (hours:minutes:seconds).
<b>GFSA</b>	Component identifier for the NFS server.
<b>348</b>	A unique message number.
<b>I</b>	Message type:  <b>A</b> Action; the user must perform a specific action. <b>E</b> Eventual action; the user must perform an action when time is available. <b>I</b> Informational; no user action is required.
<b>(I)</b>	The message level: E (error), W (attention), or I (informational). The system programmer can use the message level to determine which type of messages are shown by specifying log=error, log=warn, and log=info levels.
<b>GFSAMAIN ANMAI 02 NFS_INIT:</b>	Programming support information.
<b>OS/390 VERSION 1 NETWORK FILE SYSTEM SERVER (HDZ11TS) STARTED.</b>	Message text.

The messages are listed in numerical order (the time stamp, message level, and programming support information are not shown).

**Notes:**

1. *h\_digits* denotes hexadecimal digits, and *d\_digits* denotes decimal digits. *text* represents variable text (such as a data set name).
2. Messages GFSA300I through GFSA319I are intended for IBM support personnel when they are performing diagnosis.
3. For messages written to the console, the name of the start procedure is substituted for <PROCNAME>.

---

**GFSA320I(<PROCNAME>) NETWORK FILE  
SYSTEM SERVER INITIALIZATION  
FAILED: *text***


---

**Explanation:** *text* can be:

- VIRTUAL STORAGE IS UNAVAILABLE
- MAPPING SIDE FILE NOT FOUND
- MAPPING SIDE FILE HAS INVALID SYNTAX OR FORMAT
- ERROR OPENING/READING MAPPING SIDE FILE
- SIDE FILE SPECIFIED BUT MAPPING IS DIS-ALLOWED BY INSTALLATION
- TASK IS NOT APF AUTHORIZED
- DFP LEVEL MUST BE DFSMS 1.2 OR HIGHER
- SERVER ALREADY STARTED

**System Action:** The NFS startup ends.

**Operator Response:** Notify the system programmer.

**System Programmer Response:** The system programmer response is:

- If it is a virtual storage problem, increase the region size.
- If an APF-authorization problem, APF authorize all libraries in the STEPLIB DD statement.
- If mapping side file is not found, make sure the name specified in the attribute data set is correct and the file exists.
- If mapping side file has invalid syntax, check the part GFSAPMAP in SYS1.NFSSAMP library for mapping side file rules.
- If error occurs during opening of the side file, check to make sure that the side file is not migrated and it is readable.
- If sfmax=0 then the side file can not be specified in the attribute data set.
- If a back-level release of DFP, restart NFS server after installing DFSMSdpm™ 1.2 or a later release.
- If the same start procedure is used with the same TCP/IP stack name, use a different name for either the start procedure or the TCP/IP stack name.

---

**GFSA321I(<PROCNAME>) NETWORK FILE  
SYSTEM SERVER INITIALIZATION  
FAILED: OS/390 UNIX ADDRESS  
SPACE HAS NOT BEEN STARTED.**


---

**Explanation:** The NFS was not able to establish successful communication with the UNIX address space.

**System Action:** The NFS ends.

**Operator Response:** Before starting the NFS, both the UNIX and the TCP/IP address spaces must have successfully been started.

---

**GFSA322I(<PROCNAME>) OPENEDITION V\_REG  
FAILED: RV=1, RC=*h\_digit1*,  
RSN=*h\_digit2*.**


---

**Explanation:** The NFS failed to register.

**System Action:** The NFS ends.

**Operator Response:** Contact the system programmer.

**System Programmer Response:** *h\_digit1*, *h\_digit2* are the return code and reason code from OpenEdition V\_REG callable service. See *z/OS UNIX System Services File System Interface Reference* for more information about return code and reason codes.

---

**GFSA323I OS/390 NETWORK FILE SYSTEM LOCK  
MANAGER (HDZ11TS) WILL NOT  
START BECAUSE OMVS(UID(0)) IS  
NOT DEFINED.**


---

**Explanation:** The Network Lock Manager (NLM) initialization fails because the NLM startup procedure is not defined as OMVS(UID(0)).

**System Action:** The NLM terminates.

**User Response:** Define NLM startup procedure as OMVS(UID(0)).

---

**GFSA324I OS/390 NETWORK FILE SYSTEM  
STATUS MONITOR (HDZ11TS) WILL  
NOT START BECAUSE OMVS(UID(0))  
IS NOT DEFINED.**


---

**Explanation:** The Network Status Monitor (NSM) initialization fails because the NSM startup procedure is not defined as OMVS(UID(0)).

**System Action:** The NSM terminates.

**User Response:** Define NSM startup procedure as OMVS(UID(0)).

---

**GFS3251 REQUESTED MEMORY NOT AVAILABLE.**

**Explanation:** An operation to allocate virtual memory was tried, but was unsuccessful. If this condition persists, the cause might be:

- The value specified on the **region** parameter is too small.
- The value specified on the **bufhigh** attribute is too large.

**System Action:** The request is stopped. NFS processing continues.

---

**GFS328I(<PROCNAME>) NO SWAP REQUEST FOR NETWORK FILE SYSTEM SERVER FAILED.**

**Explanation:** The MVS NO SWAP request for NFS has failed.

**System Action:** The Network File System ends.

**Operator Response:** Contact the system programmer.

**System Programmer Response:** Try to start the NFS server again. If the failure appears to be a Network File System error, contact IBM service. Have available a symptom string and a copy of the MVS console log.

---

**GFS329I(<PROCNAME>) SERVER SHUTDOWN IN PROGRESS.**

**Explanation:** Shutdown procedures have started.

**System Action:** NFS shutdown continues.

---

**GFS330I(<PROCNAME>) SERVER SHUTDOWN COMPLETE.**

**Explanation:** Network File System and its associated subtasks have ended.

**System Action:** The Network File System stops.

---

**GFS331E(<PROCNAME>) RECALL FAILED FOR MIGRATED DATA SET *text*.**

**Explanation:** DFSMSHsm was unable to recall a data set, *text*, because the data movement program DFSMSDss™ detected, during restore, that the migrated data set had internal errors.

**System Action:** The NFS server processing continues.

**Operator Response:** Notify system programmer for recovery actions.

**System Programmer Response:** The data set *text*

had an internal error when migrated, and cannot be recalled. See DFSMSHsm Message ARC0075E for appropriate recovery actions.

---

**GFS333I(<PROCNAME>) OS/390 version NETWORK FILE SYSTEM SERVER *fmid* IS NOT STARTED BY A START COMMAND.**

**Explanation:** The system programmer tried to start the Network File System with a command other than **start**.

In the message text:

*version*      The version number, for example: 1.1.0 or VERSION 1.

*fmid*          The NFS server FMID started, for example: null or (HDZ11TS).

**System Action:** The Network File System stops.

**System Programmer Response:** Start the Network File System by issuing a **start** command.

---

**GFS334I(<PROCNAME>) MOUNT HANDLE DATABASE CANNOT BE READ.**

**Explanation:** The Network File System has tried to read the mount handle database but was unable to do so. This could occur because the mount handle database couldn't be opened or because it contained a record whose length or contents were incorrect.

**System Action:** System processing continues but in a degraded mode. Some or all the directories have not been remounted.

**Operator Response:** Contact the system programmer.

**System Programmer Response:** If this is the first startup after installation of a new release, the mount handle data sets must be cleared and the server restarted, because the format of the mount handle database is not recognizable by the new release. Clients have to reboot or unmount any previously mounted directories.

---

**GFS335E(<PROCNAME>) MOUNT HANDLE DATA SET CANNOT BE WRITTEN, EXPECTING LEN *d\_digits1* REAL LEN *d\_digits2*, VSAM R15(DEC) *d\_digits3* REASON CODE(DEC) *d\_digits4* LAST OP(DEC) *d\_digits5*.**

**Explanation:** The Network File System has tried to write a mount record to the mount handle database but was unable to do so. The length of the mount record is *d\_digits1* bytes, but only *d\_digits2* bytes were written. The failing information in writing the VSAM KSDS mount handle database is the decimal return code *d\_digits3*, the decimal error code or reason code *d\_digits4*, and the code for the last operation *d\_digits5*.

**System Action:** System processing continues but in a degraded mode. Any further mounts or unmounts are likely to fail in writing the record in mount handle data set.

**Operator Response:** Contact the system programmer.

**System Programmer Response:** The mount handle data sets have probably become unusable and either need to be cleared or restored to some previous level, and then the server has to be restarted. The information for the mount point was not saved in the mount handle data set. Clients might have to unmount and mount these mounted directories when the server is restarted.

---

**GFS3351 MOUNT HANDLE DATA SET CANNOT BE WRITTEN, EXPECTING LEN *d\_digits1* REAL LEN *d\_digits2*, VSAM R15(DEC) *d\_digits3* REASON CODE(DEC) *d\_digits4* LAST OP(DEC) *d\_digits5*.**

**Explanation:** The Network File System has tried to write a mount record to the mount handle database but was unable to do so. The length of the mount record is *d\_digits1* bytes, but only *d\_digits2* bytes were written. The failing information in writing the VSAM KSDS mount handle database is the decimal return code *d\_digits3*, the decimal error code or reason code *d\_digits4*, and the code for the last operation *d\_digits5*.

**System Action:** System processing continues but in a degraded mode. Any further mounts or unmounts are likely to fail in writing the record in mount handle data set.

**Operator Response:** Contact the system programmer.

**System Programmer Response:** The mount handle data sets have probably become unusable and either need to be cleared or restored to some previous level, and then the server has to be restarted. Clients might have to unmount and mount any previously mounted directories.

---

**GFS336E(<PROCNAME>) MOUNT HANDLE DATA SET CANNOT BE OPENED, VSAM R15(DEC) *d\_digits1* REASON CODE(DEC) *d\_digits2* LAST OP(DEC) *d\_digits3*.**

**Operator Response:** Contact the system programmer.

**System Programmer Response:** The mount handle data sets probably created with incorrect attributes.

**System Action:** The NFS stops.

**Explanation:** During resource timeout, the NFS had tried to open the mount handle data set for writing but unable to do so. The failing information in fopen the virtual storage access method (VSAM ) key-sequenced data set (KSDS) mount handle data set is the decimal return code *d\_digits1*, the decimal error code or reason

code *d\_digits2*, and the code for the last operation *d\_digits3*.

---

**GFS336I OS/390 NETWORK FILE SYSTEM LOCK MANAGER SHUTDOWN COMPLETE**

**Explanation:** The NFS Network Lock Manager (NLM) shutdown completed successfully.

**Module:** NFS Network Lock Manager (NLM).

**System Action:** NFS Network Lock Manager (NLM) ended.

---

**GFS337I THE FILE SIZE IS TOO LARGE. FN=*text*, OFFSET=*h\_digit1*, LEN=*h\_digit2*.**

**Explanation:** The Network File System client user tried to read or write the file *text* beyond the 4G.

**System Action:** The request is stopped. An error, NFSERR\_FBIG(27), is returned to the client. NFS processing continues.

**User Response:** Check the file size used. Split into multiple files and try again.

---

**GFS338I OS/390 NETWORK FILE SYSTEM STATUS MONITOR SHUTDOWN COMPLETE**

**Explanation:** The NFS Network Status Monitor (NSM) shutdown completed successfully.

**Module:** NFS Network Status Monitor (NSM).

**System Action:** NFS Network Status Monitor (NSM) ended.

---

**GFS344I OS/390 NETWORK FILE SYSTEM LOCK MANAGER (*fmid*) STARTED**

**Explanation:** The NFS Network Lock Manager (NLM) is initialized successfully and is started.

**Module:** The NFS Network Lock Manager (NLM).

**System Action:** The NFS Network Lock Manager (NLM) startup ends and continues processing.

---

**GFS345I OS/390 NETWORK FILE SYSTEM STATUS MONITOR (*fmid*) STARTED**

**Explanation:** The NFS Network Status Manager (NSM) is initialized successfully.

**Module:** The NFS Network Status Manager (NSM).

**System Action:** The NFS Network Status Manager (NSM) startup ends and continues processing.

---



---

**GFSA346I**     *time\_stamp.*

**Explanation:** Displays the current time stamp. This message is issued when the NFSLOG switches.

---

**GFSA347I**     **ERROR RETURNED TO CLIENT: RC =**  
                  *d\_digits <text>.*

**Explanation:** The error code *d\_digits* was returned to the client. *text* is the meaning of the error code.

**System Action:** Network File System processing continues.

---

**GFSA348I(<PROCNAME>) OS/390 *version***  
                  **NETWORK FILE SYSTEM SERVER *fmid***  
                  **STARTED.**

**Explanation:** The Network File System is initialized and ready to accept **modify** commands from the operator console.

In the message text:

*version*        The version number, for example: 1.1.0 or  
                  VERSION 1.

*fmid*            The NFS server FMID started, for example:  
                  null or (HDZ11TS).

**System Action:** Network File System continues processing.

---

**GFSA349I**     **UNEXPECTED ERROR DETECTED:**  
                  *d\_digits text.*

**Explanation:** The Network File System has encountered a condition that indicates continued processing might produce undesirable results. *text* is additional debugging information for the programming support personnel.

**System Action:** The Network File System either shuts down or stops the request and continue processing, depending on where the error was detected.

**System Programmer Response:** Contact your programming support personnel.

---

**GFSA360I**     *text.*

**Explanation:** This message displays memory management statistics, *text*.

**System Action:** The Network File System continues processing.

---

**GFSA361I**     **NETWORK FILE SYSTEM IS SHORT**  
                  **ON STORAGE.**

**Explanation:** This message is displayed to the operator console when a shortage of virtual storage is detected.

**System Action:** The Network File System continues processing. The storage constraint might be relieved when some storage is freed up later on.

**Operator Response:** If this message is displayed repeatedly within a short period of time, stop or cancel the Network File System and notify the system programmer.

**System Programmer Response:** Do one or both of the following before restarting the Network File System:

1. Increase the region size for the step or started task.
2. Decrease the value specified for the *bufhigh* attribute of the attributes data set.

---

**GFSA362I(<PROCNAME>) REGION SIZE WILL NOT**  
                  **ACCOMMODATE BUFHIGH AND**  
                  **LOGICAL CACHE SPECIFICATIONS.**

**Explanation:** The values of *bufhigh* and *logicalcache* specified in the attributes data set are incorrect.

**System Action:** The Network File System startup stops.

**System Programmer Response:** Either increase the region size of the step, or reduce the total of the *bufhigh* and *logicalcache* attribute values.

---

**GFSA400I**     **INVALID RECFM SPECIFICATION (*text*).**

**Explanation:** *text* is the incorrect record format specified in the attributes data set.

**System Action:** The Network File System stops.

**System Programmer Response:** Correct the attributes data set.

---

**GFSA401I(<PROCNAME>) CANNOT OPEN THE**  
                  **ATTRIBUTE DATA SET.**

**Explanation:** The server was unable to open the attributes data set defined in the JCL for DDNAME NFSATTR. The DD statement might be missing or the data set name might be incorrect.

**System Action:** The Network File System stops.

**Operator Response:** Notify your system programmer.

**System Programmer Response:** Correct the JCL for DDNAME NFSATTR.

---

---

**GFS402I(<PROCNAME>) READ FAILED FOR THE ATTRIBUTES DATA SET.**

**Explanation:** An error occurred while the Network File System was processing the attribute data set. This message follows other messages that describe the error in greater detail. The attributes data set is defined in the JCL for DDNAME NFSATTR.

**System Action:** The Network File System stops.

**Operator Response:** Notify your system programmer.

**System Programmer Response:** Correct the attributes data set. **Note:** When the NFS attribute data set is created, the *num off* option in ISPF should also be used.

The sequence number is not allowed in the NFS attributes data set.

---

**GFS403I(<PROCNAME>) PARSE FAILED IN LINE *d\_digits* text.**

**Explanation:** The parsing of line number *d\_digits* in the attribute data set failed. *text* is the actual line from the attributes data set that contains the failure. This message follows other messages that describe the error in greater detail.

**System Action:** The Network File System stops.

**System Programmer Response:** Correct the attributes data set.

---

**GFS404I UNEXPECTED END OF STRING ON END OF PARSE IN LINE *d\_digits*.**

**Explanation:** A comma is missing between attributes on line number *d\_digits* of the attributes data set.

**System Action:** The Network File System stops.

**System Programmer Response:** Correct the attributes data set.

---

**GFS405I PARSE FAILED FOR ATTRIBUTE FIELD - ILLEGAL KEYWORD IN LINE *d\_digits*.**

**Explanation:** The keyword specified in line number *d\_digits* of the attribute data set is not a valid attribute keyword.

**System Action:** The Network File System stops.

**System Programmer Response:** Correct the attributes data set.

---

**GFS406I MISSING LEFT PARENTHESIS IN LINE *d\_digits*.**

**Explanation:** An attribute specified on line number *d\_digits* of the attributes data set is missing a left parenthesis.

**System Action:** The Network File System stops.

**System Programmer Response:** Correct the attributes data set.

---

**GFS407I MISSING RIGHT PARENTHESIS IN LINE *d\_digits*.**

**Explanation:** An attribute specified on line number *d\_digits* of the attributes data set is missing a right parenthesis.

**System Action:** The Network File System stops.

**System Programmer Response:** Correct the attributes data set.

---

**GFS408I PARSE FAILED ON NUMBER FIELD IN LINE *d\_digits*.**

**Explanation:** An attribute with a negative number was specified on line number *d\_digits* of the attributes data set.

**System Action:** The Network File System stops.

**System Programmer Response:** Correct the attributes data set.

---

**GFS409I INVALID DSORG SPECIFICATION IN LINE *d\_digits*.**

**Explanation:** The data set organization specified in the *dsorg* attribute on line number *d\_digits* of the attributes data set is incorrect or is not supported by the Network File System.

**System Action:** The Network File System stops.

**System Programmer Response:** Correct the attributes data set.

---

**GFS411I INVALID VOL SPECIFICATION IN LINE *d\_digits*.**

**Explanation:** The volume specified in the *vol* (or volume) attribute on line number *d\_digits* of the attributes data set is incorrect.

**System Action:** The Network File System stops.

**System Programmer Response:** Correct the attributes data set.

---

**GFS412I INVALID UNIT SPECIFICATION IN LINE *d\_digits*.**

**Explanation:** The unit specified in the unit attribute on line number *d\_digits* of the attributes data set is incorrect.

**System Action:** The Network File System stops.

**System Programmer Response:** Correct the attributes data set.

---

**GFSA413I** THE ATTRIBUTE VALUE *d\_digits1* IS NOT IN THE RANGE OF *d\_digits2* TO *d\_digits3*.

**Explanation:** The value, *d\_digits1*, specified in one of the attributes of the attributes data set must be between the minimum value, *d\_digits2*, and the maximum value, *d\_digits3*, for this attribute. Message GFSA403I follows this message.

**System Action:** The Network File System stops.

**System Programmer Response:** See message GFSA403I to determine the attribute in error, then correct the attributes data set.

---

**GFSA414I** THE ATTRIBUTE VALUE *d\_digits1* EXCEEDS THE MAXIMUM TIMEOUT VALUE OF *d\_digits2*.

**Explanation:** The value, *d\_digits1*, specified in one of the attributes of the attributes data set must be less than or equal to *d\_digits2*. *d\_digits2* is the maximum value allowed for the attribute. Message GFSA403I follows this message.

**System Action:** Network File System stops.

**System Programmer Response:** See message GFSA403I to determine the attribute in error, then correct the attributes data set.

---

**GFSA415I** THE ATTRIBUTE TIME OUT VALUE *d\_digits1* IS LESS THAN THE MINIMUM TIME OUT VALUE *d\_digits2*.

**Explanation:** The value, *d\_digits1*, specified in the *attrtimeout* attribute in the attributes data set must be greater than or equal to the value *d\_digits2* which is specified in the *mintimeout* attribute of the attribute data set.

**System Action:** The Network File System stops.

**System Programmer Response:** Correct the attributes data set.

---

**GFSA416I** THE READ TIME OUT VALUE *d\_digits1* IS LESS THAN THE MINIMUM TIME OUT VALUE *d\_digits2*.

**Explanation:** The value, *d\_digits1*, specified in the *readtimeout* attribute of the attributes data set must be greater than or equal to the value *d\_digits2* which was specified in the *mintimeout* attribute of the attribute data set.

**System Action:** The Network File System stops.

**System Programmer Response:** Correct the attributes data set.

---

**GFSA417I** THE WRITE TIME OUT VALUE *d\_digits1* IS LESS THAN THE MINIMUM TIME OUT VALUE *d\_digits2*.

**Explanation:** The value, *d\_digits1*, specified in the *writetimeout* attribute of the attributes data set must be greater than or equal to the value *d\_digits2* which was specified in the *mintimeout* attribute of the attributes data set.

**System Action:** The Network File System stops.

**System Programmer Response:** Correct the attributes data set.

---

**GFSA418I** THE ATTRIBUTE TIME OUT VALUE *d\_digits1* IS GREATER THAN THE MAXIMUM TIME OUT VALUE *d\_digits2*.

**Explanation:** The value, *d\_digits1*, specified in the *attrtimeout* attribute of the attributes data set must be less than or equal to the value *d\_digits2* which was specified in the *maxtimeout* attribute of the attributes data set.

**System Action:** The Network File System stops.

**System Programmer Response:** Correct the attributes data set.

---

**GFSA419I** THE READ TIME OUT VALUE *d\_digits1* IS GREATER THAN THE MAXIMUM TIME OUT VALUE *d\_digits2*.

**Explanation:** The value, *d\_digits1*, specified in the *readtimeout* attribute of the attributes data set must be less than or equal to the value *d\_digits2* which was specified in the *maxtimeout* attribute of the attributes data set.

**System Action:** The Network File System stops.

**System Programmer Response:** Correct the attributes data set.

---

**GFSA420I** THE WRITE TIME OUT VALUE *d\_digits1* IS GREATER THAN THE MAXIMUM TIME OUT VALUE *d\_digits2*.

**Explanation:** The value, *d\_digits1*, specified in the *writetimeout* attribute of the attributes data set must be less than or equal to the value *d\_digits2* which was specified in the *maxtimeout* attribute of the attributes data set.

**System Action:** The Network File System stops.

**System Programmer Response:** Correct the attributes data set.

---

**GFSA421I THE NOATTRTIMEOUT ATTRIBUTE WAS SPECIFIED BUT THE MAXTIMEOUT VALUE WAS SET TO *d\_digits*.**

**Explanation:** The *noattrtimeout* attribute means that the data set is not to be deallocated following a **lookup** or **getattr** operation. The *maxtimeout* attribute specifies the maximum time out value allowed for any of the timeout attributes. *d\_digits* was specified as the maxtimeout value, in seconds, that the data set is to remain allocated. These attributes are in conflict. The *noattrtimeout* and *maxtimeout* attributes are specified in the attributes data set.

**System Action:** The Network File System stops.

**System Programmer Response:** If the *noattrtimeout* attribute is desired, the *nomaxtimeout* attribute must be specified in the attribute data set. Correct the attributes data set.

---

**GFSA422I THE NOREADTIMEOUT ATTRIBUTE WAS SPECIFIED BUT THE MAXTIMEOUT VALUE WAS SET TO *d\_digits*.**

**Explanation:** The *noreadtimeout* attribute means that the data set is not to be deallocated following a read operation. The *maxtimeout* attribute specifies the maximum time out value allowed for any of the timeout attributes. *d\_digits* was specified as the value of the *maxtimeout* attribute, in seconds, that the data set is to remain allocated. These attributes are in conflict. The *noreadtimeout* and *maxtimeout* attributes are specified in the attributes data set.

**System Action:** The Network File System stops.

**System Programmer Response:** If the *noreadtimeout* attribute is desired, the *nomaxtimeout* attribute must be specified in the attribute data set. Correct the attributes data set.

---

**GFSA423I THE NOWRITETIMEOUT ATTRIBUTE WAS SPECIFIED BUT THE MAXTIMEOUT VALUE WAS SET TO *d\_digits*.**

**Explanation:** The *nowritetimeout* attribute means that the data set is not to be deallocated following a write operation. The *maxtimeout* attribute specifies the maximum time out value allowed for any of the timeout attributes. *d\_digits* was specified as the value of the *maxtimeout* attribute, in seconds, that the data set is to remain allocated. These attributes are in conflict. The *nowritetimeout* and *maxtimeout* attributes are specified in the attributes data set.

**System Action:** The Network File System stops.

**System Programmer Response:** If the *nowritetimeout* attribute is desired, the *nomaxtimeout* attribute must be specified in the attribute data set.

Correct the attributes data set.

---

**GFSA424I MINIMUM TIME OUT VALUE, *d\_digits1*, IS GREATER THAN THE MAXIMUM TIME OUT VALUE, *d\_digits2*.**

**Explanation:** The value, *d\_digits1*, specified in the *mintimeout* attribute of the attributes data set is greater than the value, *d\_digits2*, specified in the *maxtimeout* attribute of the attributes data set.

**System Action:** The Network File System stops.

**System Programmer Response:** Correct the attributes data set.

---

**GFSA425I INVALID SPECIFICATION RECFM(*text*).**

**Explanation:** One of the characters in *text* specified on the *recfm* attribute of the attributes data set is incorrect.

**System Action:** The Network File System stops.

**System Programmer Response:** Correct the attributes data set.

---

**GFSA426I INVALID RECFM(*text*) - MUST SPECIFY U, F, OR V.**

**Explanation:** One of the characters in *text* specified on the *recfm* attribute of the attributes data set must define whether the records are fixed length (F), variable length (V), or undefined (U) format records.

**System Action:** The Network File System stops.

**System Programmer Response:** Correct the attributes data set.

---

**GFSA429I INVALID DSNTYPE SPECIFICATION IN LINE *d\_digits*.**

**Explanation:** The data set name type specified in the *dsntype* attribute on line number *d\_digits* of the attributes data set is incorrect.

**System Action:** The Network File System stops.

**System Programmer Response:** Correct the attributes data set.

---

**GFSA430I INVALID SMS\_keyword SPECIFICATION IN LINE *d\_digits*.**

**Explanation:** The SMS keyword *SMS\_keyword* shown is syntactically incorrect on line number *d\_digits*. See system-managed storage documentation for DATACLAS, MGMTCLAS, and STORCLAS naming conventions.

**System Action:** Network File System startup ends if the keyword was specified as a site attribute. If the incorrect SMS keyword was specified by a client as a mount parameter or on a command, the line number is set to zero and an I/O error is returned to the client.

**System Programmer Response:** Correct the site attributes file, if applicable.

---

**GFS4311 INVALID OPTION SPECIFICATION OF *text* IN LINE *d\_digits*.**

**Explanation:** The option specified in the **text** attribute on line number *d\_digits* of the attributes data set is incorrect.

**System Action:** Network File System stops.

**System Programmer Response:** Correct the site attributes file, if applicable.

---

**GFS4341(<PROCNAME>) *text1(d\_digits1)* IS SET TO THE DEFAULT VALUE, *text1(d\_digits2,d\_digits3,d\_digits4)*.**

**Explanation:** *text1(d\_digits1)* is the value from the previous release. This value could not be applicable to the new release. For forward compatibility, this value takes on the default value for the new release.

**System Action:** Network File System processing continues.

---

**GFS435E SUM OF *d\_digit1* PLUS *d\_digits2* MUST BE LESS THAN OR EQUAL TO *d\_digit3*.**

**Explanation:** The sum of *d\_digit1* plus *d\_digits2* is greater than *d\_digit3*.

**System Action:** Network File System processing stops.

**System Programmer Response:** Correct the values.

---

**GFS436I INVALID SIDEFILE SPECIFICATION IN LINE *d\_digits***

**Explanation:** A data set name was not specified in the sidefile attribute.

**System Action:** The NFS server startup ends.

**System Programmer Response:** Correct the problem and make necessary changes in the attribute data set.

---

**GFS437I INVALID PUBLIC SPECIFICATION IN LINE *d\_digits***

**Explanation:** The parsing of the *public* keyword resulted in error for one of the following reasons:

- *public* keyword specification is syntactically incorrect
- No public pathnames have been specified
- Pathname specified is invalid
- HFS public pathname does not match HFS prefix

**System Action:** The NFS server startup ends

**Operator Response:** Correct the problem and make the necessary changes in the attributes data set.

---

**GFS438I EXPORT SPANNING PATHNAMES NOT SUPPORTED**

**Explanation:** The Export Spanning Pathnames support for a multi-component **lookup** request is not supported.

**System Action:** The request fails. NFS processing continues.

**User Response:** Construct a different pathname in which the path is not spanned.

---

**GFS439I HFS PUBLIC PATHNAME SPECIFIED BUT HFS IS NOT ENABLED**

**Explanation:** An HFS public pathname was specified in the **public** keyword but *nohfs* was also specified which disables HFS processing.

**System Action:** The NFS server startup ends.

**Operator Response:** Correct the problem and make the necessary changes in the installation table.

---

**GFS440I INVALID SECURITY SPECIFICATION IN *d\_digits***

**Explanation:** The parsing of the *security* keyword resulted in error for one of the following reasons.

- Missing first parameter
- Invalid first parameter specified

**System Action:** The NFS server startup ends.

**Operator Response:** Correct the problem and make necessary changes in the attributes data set.

---

**GFS450I CREATED TASK(*h\_digits*) - *text1* - *text2*.**

**Explanation:** The Network File System is creating the number of tasks requested in the **nfstasks** attribute of the attributes data set. This message is displayed for each task created. *h\_digits* is the TCB address. *text1* is the task name. *text2* is the module name.

**System Action:** Network File System processing continues.

---

**GFS451I DELETING TASK(*h\_digits*) - *text*.**

**Explanation:** The Network File System is deleting a task. This is in response to the **stop** operand of the **modify** command. This message is displayed for each task deleted. *h\_digits* is the TCB address. *text* is the module name.

**System Action:** Network File System shutdown continues.

---

**GFSA452I SUBTASK TERMINATED: *h\_digits*.**

**Explanation:** The Network File System is stopping a task. This is in response to the **stop** operand of the **modify** command. This message is displayed for each ended task. *h\_digits* is the TCB address.

**System Action:** Network File System shutdown continues.

---

**GFSA470I(<PROCNAME>) NETWORK FILE  
SYSTEM SERVER SVC DUMP  
REQUEST FAILED.  
REASON=*reason\_code*.**

**Explanation:** A request to write an MVS SVC dump failed. *reason\_code* is a hexadecimal value indicating the reason MVS was unable to write the dump. See the description of the **sdump** macro in *z/OS MVS Programming: Authorized Assembler Services Reference LLA-SDU* for the meaning of the reason code.

**System Action:** Abend processing continues.

**Operator Response:** If the dump could not be written due to an operational procedure and a dump is necessary to diagnose the failure, correct the procedure.

**System Programmer Response:** If the failure appears to be due to a Network File System error, contact IBM service. Have available a symptom string and a copy of the MVS console log.

---

**GFSA471I(<PROCNAME>) NETWORK FILE  
SYSTEM SERVER ESTAE EXIT  
UNABLE TO COMPLETE  
PROCESSING. ABEND=*abend\_code*.**

**Explanation:** The Network File System ESTAE exit routine abended and is unable to complete processing. *abend\_code* is set to the last abend code encountered by a secondary instance of the ESTAE exit routine.

**System Action:** ESTAE processing is stopped. The Network File System address space might end also.

**Operator Response:** Restart the Network File System address space. Notify network users of the failure.

**System Programmer Response:** Contact IBM service. Have available a symptom string, the related SYS1.LOGREC entries, any related SDUMPs, and the Network File System error trace log.

---

**GFSA480I READ/WRITE REQUEST FAILED  
BECAUSE OF ERRORS  
ENCOUNTERED IN THE CONVERSION  
SERVICES, RC=*d\_digits1*,  
RSN=*d\_digits2*.**

**Explanation:** The *READ/WRITE* request received an error condition from the conversion services while data

was being translated. See the error codes in *OS/390 Support for Unicode™: Using Conversion Services* for an explanation of the errors encountered.

**System Action:** The request failed. NFS processing continues.

**User Response:** See *OS/390 Support for Unicode™: Using Conversion Services* for information about the error encountered and correct the problem if possible.

---

**GFSA481I READ/WRITE REQUEST FAILED  
BECAUSE TEXT CONVERSION  
RESULTED IN THE LENGTHS OF  
INPUT AND OUTPUT STRINGS BEING  
DIFFERENT.**

**Explanation:** The *READ/WRITE* request failed because the translation of the text string resulted in a different length. Translations that involved length changes (such as SBCS to MBCS) are not currently supported.

**System Action:** The request failed. NFS processing continues.

**User Response:** Make the necessary changes so that text string translations do not result in different lengths.

---

**GFSA482I READ/WRITE REQUEST FAILED  
BECAUSE TRANSLATION OF TEXT IS  
NOT POSSIBLE.**

**Explanation:** A *READ/WRITE* operation is being requested for a file that is tagged with a valid Coded Character Set Identifier (CCSID) but the user has not specified a *cln\_ccsid* to be used for translation and *xlat()* keyword is also specified.

**System Action:** The request failed. NFS processing continues.

**User Response:** Either specify a *cln\_ccsid* for translation or remove the specification *xlat()* keyword.

---

**GFSA483I WRITE REQUEST FAILED BECAUSE  
BINARY DATA CANNOT BE WRITTEN  
TO A FILE WITH PURE TEXT DATA.**

**Explanation:** A *WRITE* request has binary option specified explicitly and the file being written to has pure text data.

**System Action:** The request failed. NFS processing continues.

**User Response:** Either remove the binary option or change the file tag that is associated with the file to show that the file has mixed data.

---

---

**GFSA501I REQUEST HEADER ALLOCATION FAILED.**

---

**Explanation:** An operation to allocate virtual memory for a request header was tried, but was unsuccessful.

**System Action:** The request is stopped. Network File System processing continues.

**System Programmer Response:** Increase the size of the step region.

---

**GFSA502I REQUEST HEADER DATA BLOCK ALLOCATION FAILED.**

---

**Explanation:** An operation to allocate virtual memory for a request header data block was tried, but was unsuccessful.

**System Action:** The request is stopped. Network File System processing continues.

**System Programmer Response:** Increase the size of the step region.

---

**GFSA554I(<PROCNAME>) REGISTER FOR PORT *d\_digits1* PROGRAM *d\_digits2* - VERSION *d\_digits3* - FAILED.**

---

**Explanation:** *d\_digits1* is the port number. *d\_digits2* is the remote procedure call (RPC) program number. *d\_digits3* is the RPC program version number.

**System Action:** The Network File System stops.

**Operator Response:** Check your MVS TCP/IP setup or notify your system programmer.

**System Programmer Response:** Check your MVS TCP/IP setup.

---

**GFSA555I REGISTER FOR PORT *d\_digits1* PROGRAM *d\_digits2* - VERSION *d\_digits3* - SUCCESSFUL.**

---

**Explanation:** Information only. *d\_digits1* is the port number. *d\_digits2* is the remote procedure call (RPC) program number. *d\_digits3* is the RPC program version number.

**System Action:** The Network File System continues.

---

**GFSA556I(<PROCNAME>) UNREGISTER PROGRAM *d\_digits1* VERSION *d\_digits2* - FAILED.**

---

**Explanation:** *d\_digits1* is the remote procedure call (RPC) program number. *d\_digits2* is the RPC program version number.

**System Action:** Network File System stops.

**Operator Response:** Contact system programmer.

**System Programmer Response:** This can be caused

by an out-of-memory error. Increase the size of the job and step region.

---

**GFSA557I UNREGISTER PROGRAM *d\_digits1* VERSION *d\_digits2* - SUCCESSFUL.**

---

**Explanation:** Information only. *d\_digits1* is the remote procedure call (RPC) program number.

**System Action:** Network File System processing continues.

---

**GFSA558I(<PROCNAME>) UNABLE TO CREATE IPC QUEUE.**

---

**Explanation:** An operation to allocate virtual memory for an inter-process communication (IPC) queue was tried, but was unsuccessful.

**System Action:** The Network File System stops.

**Operator Response:** Notify your system programmer.

**System Programmer Response:** Increase the size of the step region.

---

**GFSA559I(<PROCNAME>) CANNOT CREATE UDP SERVICE.**

---

**Explanation:** User datagram protocol (UDP) service transport could not be created, or you started TCP/IP before the UNIX initialization was completed.

**System Action:** The Network File System stops.

**Operator Response:** Check your MVS TCP/IP setup or notify your system programmer. Before starting TCP/IP, make sure that the UNIX initialization has completed, and the TCP/IP UNIX connection is established.

**System Programmer Response:** Check your MVS TCP/IP setup. Also check your UNIX BPXPRMxx parmlib member, specifically the MAXFILEPROC, MAXSOCKETS, INADDRANYPORT, and INADDRANYCOUNT. The INADDRANYPORT and INADDRANYCOUNT must be specified, but its range can not include 2049, in order for the server to initialize. See *z/OS MVS Initialization and Tuning Guide* for more details.

---

**GFSA564I(<PROCNAME>) NETWORK SERVICE IS UNAVAILABLE.**

---

**Explanation:** If the **Portmapper** or TCP/IP is not running, the Network File System fails to initialize and the startup ends.

**System Action:** The Network File System startup ends.

**System Programmer Response:** Ensure that both the **Portmapper** and TCP/IP are operational.

---

**GFSA565I(<PROCNAME>) OS/390 NFS SERVER  
CANNOT OBTAIN NFS PORT 2049.**

**Explanation:** The Network File System cannot obtain NFS port 2049.

**System Action:** The Network File System ends.

**Operator Response:** Check your MVS TCP/IP setup or notify your system programmer.

**System Programmer Response:** Check your MVS TCP/IP setup.

---

**GFSA566I(<PROCNAME>) CANNOT ACCEPT NEW  
TCP CLIENT CONNECTION —  
MAXIMUM NUMBER OF SOCKETS HAS  
REACHED**

**Explanation:** When an NFS TCP client attempts to connect to the NFS server, the server cannot accept the connection because the maximum number of sockets has been reached.

**System Action:** The connection request fails. NFS processing continues.

**Operator Response:** Increase the value of **maxsockets** for AF\_INET domain in the BPXPRMxx parmlib member so that client TCP connections can be accepted by the NFS server. See *z/OS UNIX System Services Planning* for more information.

---

**GFSA750I(<PROCNAME>) SMF PROCESSING  
ACTIVE FOR USER LOGOUT.**

**Explanation:** SMF processing is active for user logout records. This message is displayed in response to the **status** operand on the **modify** command.

**System Action:** NFS server processing continues.

---

**GFSA751I(<PROCNAME>) SMF PROCESSING  
SUSPENDED FOR USER LOGOUT.**

**Explanation:** System-managed facility (SMF) processing is suspended for user logout records. This message is displayed in response to the **status** operand on the **modify** command, or after receiving a non-zero return code from SMF. See message GFSA754I for SMF return code.

**System Action:** NFS server processing continues without generating any user or file SMF records.

**Operator Response:** Resume SMF recording by entering a **modify** command specifying SMF=ON.

---

**GFSA752I(<PROCNAME>) SMF PROCESSING  
ACTIVE FOR FILE TIMEOUT.**

**Explanation:** System-managed facility (SMF) processing is active for file timeout records. This message is displayed in response to the **status**

operand on the **modify** command.

**System Action:** NFS server processing continues.

---

**GFSA753I(<PROCNAME>) SMF PROCESSING  
SUSPENDED FOR FILE TIMEOUT.**

**Explanation:** System-managed facility (SMF) processing is suspended for file timeout records. This message is displayed in response to the status operand on the **modify** command, or after receiving a non-zero return code from SMF. See message GFSA754I for SMF return code.

**System Action:** NFS server processing continues without generating any user or file SMF records.

**Operator Response:** Resume SMF recording by entering a **modify** command specifying SMF=ON.

---

**GFSA754I(<PROCNAME>) UNEXPECTED RETURN  
CODE *d\_digits* RECEIVED FROM SMF  
WHILE WRITING RECORD TYPE 42  
SUBTYPE [7|8].**

**Explanation:** The NFS server received a non-zero return code *d\_digits* while processing a file timeout (subtype 7) or user logout (subtype 8) record.

**System Action:** Network File System processing continues. No more SMF records of the same type and subtype are generated until the Network File System address space is restarted or the SMF=ON operand of the **modify** command is entered.

**Operator Response:** Notify the system programmer.

**System Programmer Response:** See *z/OS MVS System Management Facilities (SMF)* for information about the return code. Correct the problem, and have the operator enter a **modify** command specifying SMF=ON.

---

**GFSA770I OS/390 UNIX REGISTRATION  
SUCCESSFUL.**

**Explanation:** Connection with UNIX established.

**System Action:** Network File System processing continues.

---

**GFSA771I(<PROCNAME>) HFS MOUNTS  
SUSPENDED.**

**Explanation:** Hierarchical file system (HFS) mount processing has been suspended by the **freeze=onhfs** operand of the **modify** command. This message is displayed in response to either the **freeze=onhfs** or **status** operands of the **modify** command. Further HFS mount requests from the network are ignored. Existing mounts are unaffected.

**System Action:** Network File System processing continues.



---

**GFSA772I(<PROCNAME>) HFS MOUNTS  
RESUMED.**

**Explanation:** Mount requests to hierarchical file system (HFS) file systems have been enabled. Given in response to a console **modify** command.

**System Action:** Network File System processing continues.

---

**GFSA776I HFS CANNOT RESOLVE PATH NAME  
text.**

**Explanation:** There was a failure to resolve a path name with UNIX when initializing from the mount handle data sets. The HFS file system has been removed or renamed. If the user attempts to access a file object under this mount point, the Network File System error response NFSERR\_STALE is returned.

**System Action:** Network File System processing continues.

---

**GFSA777I SERVICE REQUESTER DOES NOT  
HAVE SECURITY PRIVILEGE.**

**Explanation:** The client user must be defined to Resource Access Control Facility (RACF) as a user of OpenEdition multiple virtual system (MVS) to access hierarchical file system (HFS) file objects.

**System Action:** The request is stopped. Network File System processing continues.

**System Programmer Response:** Check the System Authorization Facility (SAF) security product user profiles.

---

**GFSA782I(<PROCNAME>) NO ACTIVE HFS MOUNT  
POINTS.**

**Explanation:** This message is in response to the **list=mounts** operand of the **modify** command and shows that there are no OpenEdition clients connected to the Network File System.

**System Action:** Network File System processing continues.

---

**GFSA783I(<PROCNAME>) NO ACTIVE HFS DATA  
SETS.**

**Explanation:** This message is in response to the **list=dsnames** operand of the **modify** command and shows that there are no clients actively accessing hierarchical file system (HFS) data sets.

**System Action:** Network File System processing continues.

---

**GFSA784I text1 RPC OS/390 UNIX ERROR  
VNODE\_OP text2 RC: number1: text3  
RSN: number2 number3.**

**Explanation:** The NFS has encountered a problem on an interface call with UNIX. The error was encountered during processing of a *text1* remote procedure call (RPC) call. *text2* is the function called when failure occurred. *number1* is the return code and *text3* is the English description of the return code. The combination of *number2 number3* represents the reason code as returned by UNIX. See UNIX documentation for a full explanation of OpenEdition MVS reason codes (for example, *z/OS UNIX System Services Programming: Assembler Callable Services Reference*, SA22-7803).

**System Action:** The NFS continues processing.

**System Programmer Response:** See the UNIX documentation.

---

**GFSA786I MULTI-COMPONENT LOOKUP  
REQUEST FOR PATHNAME text  
CANNOT BE RESOLVED.**

**Explanation:** The multi-component **lookup** request for the path name failed. Possible reasons are:

- Symbolic links cannot be embedded in a multi-component path name as it is not supported at this time
- Pathname specified is either not supported or access is not allowed
- Public path is not set up on this server.

**System Action:** The request fails. NFS processing continues.

**User Response:** Construct a different **lookup** request with a valid path name.

---

**GFSA801I MOUNT FAILED: text**

**Explanation:** The text can be:

- FILE MAPPING ENABLED BUT NO SIDE FILE SPECIFIED
- STORAGE LIMIT REACHED LOADING MAPPING SIDE FILE
- MAPPING SIDE FILE NOT FOUND
- ERROR OPENING/READING MAPPING SIDE FILE
- MAPPING SIDE FILE HAS INVALID SYNTAX OR FORMAT
- SIDE FILE SPECIFIED BUT MAPPING IS DIS-ALLOWED BY INSTALLATION

**User Response:**

1. Specify a side file if **fileextmap** is ON, or
2. Fix problem with mapping side file, or
3. Consult the system administrator to change **sfmax** value and re-issue the **mount** command.

---

**GFSA802E REMOUNT FAILED — PHYSICAL FILE SYSTEM CHANGED. PATH: pathname PREV: datasetname CURRENT: datasetname**

**Explanation:** During restart of the NFS server, rebuild of the mount point recorded in the Mount Handle Database failed. The failure occurred because the physical file system for the mount point was changed by a **tso unmount** since the mount point was originally mounted.

**System Action:** NFS server processing continues.

---

**GFSA811I CANNOT FLUSH PARTIAL RECORDS FOR DATA SET text1(text2): FB h\_digits.**

**Explanation:** There was not enough memory to allocate storage required to flush partial records to the data set at data set close time. The partial record was discarded.

**System Action:** The data set is closed. Network File System processing continues.

**System Programmer Response:** Increase the size of the step region. The data set might be incomplete.

---

**GFSA812I FLUSH FAILED: RC d\_digits1 OFFSET d\_digits2 WAS DROPPED FOR DATA SET text1(text2).**

**Explanation:** When the Network File System was trying to flush cached data at data set close time, the error *d\_digits1* was detected. The data at offset *d\_digits2* was discarded. This error message follows more specific error messages.

**System Action:** The data set is closed. Network File System processing continues.

**System Programmer Response:** See the message preceding this message to determine the correct action. The data set might be incomplete. See Table 38 on page 139 for a description of the return code *d\_digits1*.

---

**GFSA813I REMOVE FAILED: RC h\_digits DSN text1(text2).**

**Explanation:** The error *h\_digits* was detected when trying to remove member *text2* from PDS *text1*.

**System Action:** Network File System processing continues.

**System Programmer Response:** *h\_digits* is the return code from the MVS/DFP STOW macro. See *z/OS DFSMS Macro Instructions for Data Sets* for the explanation of STOW return codes.

---

**GFSA814I RENAME FAILED: RC h\_digits DSN text1 OLDMEM text2 NEWMEM text3.**

**Explanation:** The error *h\_digits* was detected when trying to rename member *text2* to *text3* in partitioned data set *text1*.

**System Action:** Network File System processing continues.

**System Programmer Response:** *h\_digits* is the return code from the MVS/DFP STOW macro. See *z/OS DFSMS Macro Instructions for Data Sets* for the explanation of STOW return codes.

---

**GFSA815I RENAME FAILED: RC h\_digits1 OLDDSN text1 VOL text2 UNIT h\_digits2 NEWDSN text3.**

**Explanation:** The error *h\_digits1* was detected when trying to rename the non-VSAM data set *text1* to *text3*. *text2* is disk volume serial number. *h\_digits2* is the disk device type code.

**System Action:** Network File System processing continues.

**System Programmer Response:** *h\_digits1* is a composite of one or more error return codes encountered when trying to rename the data set. The hexadecimal digits are decoded using the list below.

Renaming a data set requires three steps: Uncatalog the old data set name, rename the data set in the disk VTOC, and catalog the new data set name. Should an error occur in either of the last two steps, the prior steps are undone to preserve the old data set name.

Find the step that failed by matching the value in byte zero of the return code with the values under the heading **byte 0** below. Byte three contains the return code from the first failing MVS/DFP service (Uncatalog/catalog or DADSM rename). If further errors occur when trying to recatalog or rename the data set back to the old name, the return codes are placed in bytes one and two respectively.

#### **Byte 0 Meaning/Other Bytes**

- |           |   |
|-----------|---|
| <b>00</b> | Error uncataloging old data set name. Byte 3: Uncatalog return code.  |
| <b>01</b> | Error renaming the data set. Byte 1: Recatalog return code for old data set name. Byte 3: DADSM rename return code.   |
| <b>02</b> | Error cataloging new data set name. Byte 1: Recatalog return code for old data set name. Byte 2: DADSM rename return code for old data set name. Byte 3: Catalog return code for new data set name. |

Catalog and DADSM rename return codes are documented in *z/OS DFSMSdfp Advanced Services*. Message IEC614I is written for DADSM rename errors

and contains further diagnostic codes. These codes are documented in *z/OS DFSMSdfp Diagnosis Reference*.

---

**GFSA816I HOST NAME OF IP ADDRESS (*d\_digits*) WAS NOT FOUND BY TCP/IP.**

**Explanation:** The client host name of IP address *d\_digits* is not defined in either the TCP/IP Domain Name Server or the TCP/IP Site Table. See *TCP/IP for MVS: Customization and Administration Guide*.

**System Action:** Network File System processing continues. The dotted IP address is used as the host name.

**System Programmer Response:** Insert this client host's entry into either the TCP/IP Domain Name Server or the TCP/IP Site Table.

---

**GFSA817I *text1* REQUEST NOT VALID ON ALIAS NAME *text2*.**

**Explanation:** *text1* can be remove or rename. *text2* is an MVS access method services alias name of a file which also has a true name. Remove (**rm** or **rmdir**) and **rename** (**mv**) requests cannot be run using an alias name. The true file name is required.

**System Action:** The request is stopped. An I/O error indication is returned to the client. Network File System processing continues.

**System Programmer Response:** Inform the client user of this error.

**User Response:** Provide the true name of the file on the request.

---

**GFSA818I(<PROCNAME>) EXPORTS: NO VALID HOST NAMES IN *text* LIST.**

**Explanation:** None of the client host names in the read/write or access list (see *text*) are defined to the network.

**System Action:** The Network File System does not export the associated directory if the result is a null access list. If the result is a null read/write list, the directory is exported read-only.

**System Programmer Response:** Correct the host names in the exports data set or have the host names defined to the network.

---

**GFSA819I DATA SET *text1* CREATION USING DATA CLASS = *text2*.**

**Explanation:** Data set *text1* is being allocated using the attributes in data class *text2*.

**System Action:** Network File System processing continues.

---

**GFSA820I CATALOG ERROR OCCURRED WHILE [RETRIEVING|UPDATING] CATALOG INFORMATION FOR *text*. RETURN CODE IS *d\_digits1*, REASON CODE IS *cc-d\_digits2*.**

**Explanation:** Catalog Management module IGG0CLcc returned this return code, *d\_digits1*, and reason code, *d\_digits2*, as the result of a catalog error or an exception condition. *text* was the name of the data set that the [retrieve|update] operation was performed against.

**System Action:** Network File System processing continues.

**System Programmer Response:** See message IDC3009I in the *z/OS MVS System Messages, Vol 5*, *z/OS MVS System Messages, Vol 6*, and *z/OS MVS System Messages, Vol 7* manuals for specific return code and reason code information.

---

**GFSA821I ERROR OCCURRED WHILE UPDATING THE FORMAT 1 DSCB FOR *text1* ON *text2*. FUNCTION CODE IS *d\_digits1*, RETURN CODE IS *d\_digits2*, REASON CODE IS *d\_digits3*.**

**Explanation:** *text1* is the name of the data set and *text2* is the volume serial number of the volume on which the data set resides. The function code can be one of the following:

- 2 De-serializing the UCB
- 4 De-serializing the DASD volume
- 12 Searching for the UCB
- 16 Serializing the DASD volume
- 20 Reading the DSCB
- 24 Writing the DSCB

**System Action:** Network File System processing continues.

**System Programmer Response:** See the following manuals for specific return code and reason code information:

- 2 *z/OS MVS Programming: Authorized Assembler Services Reference SET-WTO*, macro UCBPIN.
- 4 *z/OS MVS Programming: Authorized Assembler Services Reference ALE-DYN*, macro DEQ.
- 12 *z/OS MVS Programming: Authorized Assembler Services Reference SET-WTO*, macro UCBLook.
- 16 *z/OS MVS Programming: Authorized Assembler Services Reference LLA-SDU*, macro RESERVE.

- 20      *z/OS DFSMSdfp Advanced Services*, Return Codes from CVAFDIR.
- 24      *z/OS DFSMSdfp Advanced Services*, Return Codes from CVAFDIR.

---

**GFSA822I    UNABLE TO PERFORM FILE MAPPING  
BECAUSE NO SIDE FILE SPECIFIED  
OR LOADED**

**Explanation:** File mapping cannot be performed because a side file was not specified either as a default or at the mount point.

**System Action:** Operation fails

**System Programmer Response:** Specify a side file in the attributes data set.

**User Response:** Specify a side file in the mount command.

---

**GFSA823I(<PROCNAME>) PUBLIC PATH CANNOT  
BE ESTABLISHED.**

**Explanation:** The public pathname(s) specified in the *public* keyword cannot be established during server startup. This could be because the path is not exported or it does not exist.

**System Action:** The NFS server startup ends.

**Operator Response:** Make sure the public pathname exist and is exported if export list checking is enabled. Correct the problem and restart the server.

---

**GFSA827I    REQUEST *h\_digits* CREDENTIALS  
ALLOCATION FAILED.**

**Explanation:** An operation to allocate virtual memory for a credentials block was tried, but was unsuccessful. *h\_digits* is the request block address.

**System Action:** The request is stopped. Network File System processing continues.

**System Programmer Response:** Increase the size of the step region.

---

**GFSA829I    REQUEST *h\_digits* INVALID  
CREDENTIALS FLAVOR *d\_digits*.**

**Explanation:** Incorrect credentials type received from client. Probable client software error. The Network File System supports UNIX and non-authentication styles. *h\_digits* is the request block address.

**System Action:** The request is stopped. Network File System processing continues.

**System Programmer Response:** Inform the client user that the credentials used are not valid.

---

**GFSA832I    REQUEST *h\_digits* INVALID  
MEMBERNAME FOR *text*.**

**Explanation:** *text* is the member name of a partitioned data set (PDS) that was specified as a file name by the Network File System client user. The file name specified was incorrect or not found in the PDS. *h\_digits* is the request block address.

**System Action:** The request is stopped. Network File System processing continues.

**System Programmer Response:** Inform the client user of this error.

**User Response:** Correct the error and resubmit the request.

---

**GFSA833I    REQUEST *h\_digits* PARSE FAILED FOR  
*text*.**

**Explanation:** *text* is the member name of a partitioned data set (PDS) or a data set name that was specified as a file name by the Network File System client user. The file name specified was incorrect, not found in the PDS, or was an incorrect or non-existent data set. *h\_digits* is the request block address.

**System Action:** The request is stopped. Network File System processing continues.

**System Programmer Response:** Inform the client user of this error.

**User Response:** Correct the error and resubmit the request.

---

**GFSA840I    DYNAMIC ALLOCATION FAILED WITH  
RETURN CODE *h\_digits* FOR DATA SET  
*text1(text2)*.**

**Explanation:** Dynamic file allocation error. *h\_digits* is the dynamic allocation return code. *text1* is the data set name. *text2* is the member name if any.

**System Action:** The request is stopped. Network File System processing continues.

**System Programmer Response:** This message is preceded by either message GFSA853I or GFSA854I. See the programmer response for the message that precedes this message to determine the appropriate action.

---

**GFSA842I    *text1* UNSUPPORTED DSORG *text2*.**

**Explanation:** The data set organization of data set *text1* is not supported by the Network File System. *text2* can be the characters "ISAM" or "UNKNOWN"

**System Action:** The request is stopped. Network File System processing continues.

**System Programmer Response:** Inform the client user of this error.

---

**GFSA843I CREATE FAILED FOR *text*.**

**Explanation:** An error occurred while the Network File System was trying to create the data set *text*. This message follows other messages that describe the error in greater detail.

**System Action:** The request is stopped. Network File System processing continues.

**System Programmer Response:** See the message(s) preceding this message to determine the appropriate response.

---

**GFSA847I IDCAMS ERROR: *text*.**

**Explanation:** *text* is an access method services error message.

**System Action:** The request is stopped. Network File System processing continues.

**System Programmer Response:** See *MVS System Messages* for more information on the access method services error message, or use **LookAt**. For a description of LookAt, see "Using LookAt to look up message explanations" on page xvi.

---

**GFSA848I PDS *text* IS NOT EMPTY.**

**Explanation:** The Network File System client user issued a **rmdir** (remove directory) AIX or UNIX command to remove a partitioned data set that was not empty. The Network File System version 2 protocol specification requires the directory (PDS) to be empty before it is removed. This is a Network File System client user error. *text* is the name of the PDS.

**System Action:** The request is stopped. An error is returned to the client. Network File System processing continues.

**System Programmer Response:** Inform the Network File System client user of this error.

**User Response:** Remove all files in the directory then resubmit the **rmdir** request.

---

**GFSA849I NEW AND OLD FILES ARE NOT MEMBERS OF THE SAME PDS.**

**Explanation:** Rename is not allowed for a member of a PDS when the target name is not in the same PDS. This is a Network File System client user error.

**System Action:** The request is stopped. An error is returned to the client. Network File System processing continues.

**System Programmer Response:** Inform the Network File System client user of this error.

**User Response:** Check the filename used. Correct it and try again.

---

**GFSA853I DYNAMIC ALLOCATION: INPUT VALIDATION ROUTINE REJECTED ALLOCATION.**

**Explanation:** Dynamic allocation failed when issued by the installation input validation routine. Probable installation configuration errors.

**System Action:** The request is stopped. Network File System processing continues.

**System Programmer Response:** See the *z/OS MVS Installation Exits* for more information on the Input Validation routine.

---

**GFSA854I DYNAMIC ALLOCATION: RC=*d\_digits* ERROR=*h\_digits1* INFO=*h\_digits2* : *text*.**

**Explanation:** Dynamic allocation failed with return code *d\_digits*, error reason code *h\_digits1* and information reason code *h\_digits2*. *text* is a description of the interrupted dynamic allocation request.

**System Action:** The request is stopped. Network File System processing continues.

**System Programmer Response:** See the *z/OS MVS Programming: Authorized Assembler Services Guide* for more information about these codes.

---

**GFSA858I OPEN FAILED RC *d\_digits* FOR DATA SET *text1(text2)*.**

**Explanation:** An error occurred while trying to open the data set. *text1* is the data set name. *text2* is the member name (if any). *d\_digits* is the return code.

**System Action:** The request is stopped. Network File System processing continues.

**System Programmer Response:** If this error was not caused by an out-of-memory condition, contact your programming support personnel. See Table 38 on page 139 for a description of the return code *d\_digits*.

---

**GFSA859I READ FAILED RC *d\_digits* FOR DATA SET *text1(text2)*.**

**Explanation:** An error occurred while trying to read the data set. *text1* is the data set name. *text2* is the member name (if any). *d\_digits* is the return code.

**System Action:** The request is stopped. Network File System processing continues.

**System Programmer Response:** If this error was not caused by an out-of-memory condition, contact your programming support personnel. See Table 38 on page 139 for a description of the return code *d\_digits*.

---

**GFSA860I WRITE FAILED RC *d\_digits* FOR DATA SET *text1(text2)*.**

**Explanation:** An error occurred while trying to write the data set. *text1* is the data set name. *text2* is the member name (if any). *d\_digits* is the return code.

**System Action:** The request is stopped. Network File System processing continues.

**System Programmer Response:** If this error was not caused by an out-of-memory condition, contact your programming support personnel. See Table 38 on page 139 for a description of the return code *d\_digits*.

---

**GFSA862I CATALOG (*text*) COULD NOT BE LOCATED.**

**Explanation:** The user catalog named *text* that contains the entry for an index could not be located. The catalog does not exist or is not mounted. If it does not exist, the entry in the master catalog might be incorrect.

**System Action:** Network File System processing continues.

**System Programmer Response:** Investigate why the catalog could not be found and take corrective action.

---

**GFSA863I READDIR ON ROOT IS NOT ALLOWED.**

**Explanation:** The user tried to list the contents of the master catalog.

**System Action:** The request is stopped. Network File System processing continues.

---

**GFSA864I(<PROCNAME>) CANNOT OPEN THE EXPORTS DATA SET.**

**Explanation:** The server was unable to open the exports data set defined in the job control language (JCL) for DDNAME EXPORTS. The DD statement might be missing or the data set name might be incorrect.

**System Action:** The Network File System stops.

**Operator Response:** Notify your system programmer.

**System Programmer Response:** Correct the JCL for DDNAME EXPORTS.

---

**GFSA865I(<PROCNAME>) EXPORTS: UNEXPECTED OPTION (*text*)-- SHUTDOWN SCHEDULED.**

**Explanation:** The option information provided in *text* is incorrect. This error could occur as a result of unexpected blanks, incorrect syntax, or mutually exclusive options (for example, both *ro* and *rw*).

**System Action:** Checking of the exports data set continues but shutdown of the Network File System occurs at its completion.

**System Programmer Response:** Correct the exports data set and restart the Network File System.

---

**GFSA866I(<PROCNAME>) EXPORTS: DIRECTORY *text* WAS NOT EXPORTED.**

**Explanation:** An error was encountered severe enough to prevent the data set or index named *text* from being exported. This message follows a more specific error message.

**System Action:** Network File System processing continues.

**System Programmer Response:** Correct the exports data set.

---

**GFSA867I(<PROCNAME>) EXPORTS: *text1* CANNOT BE EXPORTED BECAUSE *text2* ALREADY IS.**

**Explanation:** The data set or index named in *text1* is a parent directory or a subdirectory of the data set or index named in *text2* which is already exported.

**System Action:** Network File System processing continues.

**System Programmer Response:** Correct the exports data set.

---

**GFSA868I(<PROCNAME>) EXPORTS: HOST (*text*) WITH UNKNOWN ADDRESS TYPE (*d\_digits*) HAS BEEN IGNORED.**

**Explanation:** The host named *text* has an IP address (*d\_digits*) that is not valid.

**System Action:** Network File System processing continues.

**System Programmer Response:** Correct the exports data set.

---

**GFSA869I(<PROCNAME>) EXPORTS: UNKNOWN HOST (*text*) HAS BEEN IGNORED.**

**Explanation:** The client host named *text* is not defined to the network. If there are other defined client hosts specified for the option, the Network File System ignores this undefined client host as if had not been specified. (However, see message GFSA818I for special handling of null lists.)

**System Action:** Network File System processing continues.

**System Programmer Response:** Correct the host name in the exports data set or have this host name defined to the network.

---

**GFSA8711**    **REQUEST *h\_digits* HAS MISMATCHED  
UID: CRED = *d\_digits1* ARGS =  
*d\_digits2*.**

**Explanation:** *h\_digits* is a block created for the logon or logout request. *d\_digits1* represents the credential user ID number. *d\_digits2* represents the client user ID number. The user ID numbers do not match and this is considered a security failure.

**System Action:** The client logon or logout request is stopped. Network File System processing continues.

**System Programmer Response:** This is a Network File System client application problem. If the Network File System client application is offered by IBM, inform the IBM programming support center. If the Network File System client application is offered by your programming support personnel, contact your programming support personnel.

---

**GFSA8761**    **I/O ERROR ON DSN = *text1(text2)*  
SENSE *h\_digits1* IOBCSW *h\_digits2*  
*h\_digits3*. ACCESS METHOD RC =  
*h\_digits4* ACCESS METHOD RSN =  
*h\_digits5***

**Explanation:** The physical I/O layer tried to check some previous operation in the data set and the check failed. *text1* is the data set name. *text2* is the member name (if any). *h\_digits1* is the sense bytes 0 and 1 from the device. *h\_digits2* is the first 3 bytes of the channel status word from the device. *h\_digits3* is the last 4 bytes of the channel status word from the device. *h\_digits4* is the access method return code. *h\_digits5* is the access method reason code.

**System Action:** The request is stopped. Network File System processing continues.

**System Programmer Response:** See the appropriate device documentation for more information on the sense bytes and channel status word.

---

**GFSA8771**    **R0=*h\_digits1* R1=*h\_digits2*: *text* ACCESS  
METHOD RC = *h\_digits3* ACCESS  
METHOD RSN = *h\_digits4***

**Explanation:** A SYNAD error was detected during physical I/O operation. *h\_digits1* is the contents of register 0; *h\_digits2* is the contents of register 1. The *text* is the message returned from the SYNAD analysis function macro. *h\_digits3* is the access method return code. *h\_digits4* is the access method reason code.

**System Action:** The request is stopped. If the error detected is a B37, D37 or E37 abend, the Network File System restores the file size to the last known file size before the SYNAD error. Network File System processing continues.

**System Programmer Response:** A Data Management message should have been displayed on the console. See *MVS System Messages* for a

description of the return code to determine the corrective action or use LookAt. For a description of LookAt, see "Using LookAt to look up message explanations" on page xvi.

---

**GFSA8781(<PROCNAME>) CANNOT OPEN THE  
CHKLIST DATA SET.**

**Explanation:** The Network File System cannot open the checklist data set, specified by the CHKLIST DD statement in the MVS NFS startup procedure.

**System Action:** The Network File System ends.

**Operator Response:** Contact the system programmer.

**System Programmer Response:** Check for validity of the specified data set. Correct the JCL for CHKLIST DD statement in the MVS NFS startup procedure.

---

**GFSA8791(<PROCNAME>) PROBLEMS  
ENCOUNTERED PARSING THE  
CHKLIST DATA SET.**

**Explanation:** The Network File System was not able to successfully parse the checklist data set specified by the CHKLIST DD statement.

**System Action:** The Network File System ends.

**Operator Response:** Contact the system programmer.

**System Programmer Response:** Correct the content of the checklist data set.

---

**GFSA8811**    **FUB: *h\_digits1* UNABLE TO ACCESS  
FILE *text1(text2)* OWNED BY FUB  
*h\_digits2*.**

**Explanation:** A user has tried to access a data set already locked internally by the Network File System for writing by another user. The data set has not been released yet. *h\_digits1* is the address of the File Usage Block. *text1* is the data set name. *text2* is the member name (if any). *h\_digits2* is the address of the File Usage Block which currently has the file allocated.

**System Action:** The request is stopped. A "Not Owner" error message is returned to the user. Network File System processing continues.

**System Programmer Response:** If queried by the user, the "Not Owner" error message, as it relates to the Network File System, is described in *z/OS Network File System User's Guide*.

---

**GFSA8861**    **UNABLE TO WRITE RECNO *d\_digits* TO  
*text1(text2)* DUE TO PREVIOUS ERROR.**

**Explanation:** An error was previously detected which prevents writing to a data set.

**System Action:** The request is stopped. Network File System processing continues.

**System Programmer Response:** If the previous error cannot be determined, contact your programming support personnel.

---

**GFSA895I REQUEST *h\_digits* - FILE *text* NOT ALLOCATED.**

**Explanation:** The Network File System did not have the data set *text* open on a request to close the file. The file name might have been specified incorrectly, or the time out might have already occurred for this data set, causing the server to close the data set.

In the message text:

*h\_digits*      The request block address.

*text*          The data set name.

**System Action:** The system ends the request. Return code X'131' is passed back to the user. The system continues processing for the Network File System.

---

**GFSA896I REQUEST *h\_digits1* - FILE BLOCK *h\_digits2* ASSOCIATED WITH FILE *text* NOT IN USE BY CREDENTIALS *h\_digits3*.**

**Explanation:** The request to close file *text* was received, but the file was not opened by the client. A file can only be closed by the same client that opened the file. *h\_digits1* is the request block address. *h\_digits2* is the file block address. *h\_digits3* is the credentials block address.

**System Action:** The request is stopped. Return code 132 is passed back to the client. Network File System processing continues.

---

**GFSA897I RECORD *d\_digits1* SIZE *d\_digits2* IS *text1*: *text2* = *d\_digits3* DSN = *text3*(*text4*).**

**Explanation:** The record received from the client can be too long or too short. *d\_digits1* is the record number. *d\_digits2* is the record size. *text1* can be the characters "TOO LONG" or "TOO SHORT". *text2* can be the characters "MAXIMUM" or "MINIMUM". *d\_digits3* shows the maximum or minimum length allowed. *text3* is the data set name. *text4* is the member name (if any).

**System Action:** The request is stopped. Network File System processing continues.

---

**GFSA898I EOL SEQUENCE MISMATCH FOR DATA SET *text1*(*text2*).**

**Explanation:** The end-of-line terminator was not found in the same place as the previous end-of-line terminator for an offset that is being re-written by the client. This is a Network File System client error.

**System Action:** The request is stopped. Network File System processing continues.

**User Response:** Make sure that the record you are writing is the same size as the record you are replacing.

---

**GFSA899I BLANKSTRIP MODE: TRAILING BLANK(S) IN RECORD *d\_digits* IS NOT ALLOWED DSN = *text1*(*text2*).**

**Explanation:** Writing data in text mode with blank stripping enabled and blanks at the end of the line to a data set with fixed-length records is not allowed.

**System Action:** The request is stopped. Network File System processing continues.

---

**GFSA900I(<PROCNAME>) MOUNT PROCESSING ACTIVE.**

**Explanation:** Mounts can be issued from the network. This message is displayed in response to the **status** operand of the **modify** command.

**System Action:** Network File System processing continues.

---

**GFSA901I(<PROCNAME>) MOUNT PROCESSING SUSPENDED.**

**Explanation:** Mount processing has been suspended by the **freeze=on** operand of the **modify** command. This message is displayed in response to either the **freeze=on** or **status** operands of the **modify** command. Further mount requests from the network are ignored. Existing mounts are unaffected.

**System Action:** Network File System processing continues.

---

**GFSA902I(<PROCNAME>) MOUNT PROCESSING RESUMED.**

**Explanation:** Mount processing has been resumed by the **freeze=off** operand of the **modify** command. Further mount requests from the network are be honored. Existing mounts are unaffected.

**System Action:** Network File System processing continues.

---

**GFSA903I(<PROCNAME>) TASK *h\_digits1* TCB *h\_digits2* PROGRAM = *text1* = *text2*.**

**Explanation:** *h\_digits1* is the task queue address. *h\_digits2* is the task control block address. *text1* is the program name. *text2* is the specific task name. This is in response to the **status** operand of the **modify** command.

**System Action:** Network File System processing continues.



---

**GFSA904I(<PROCNAME>) HFS MOUNT  
PROCESSING ACTIVE.**

**Explanation:** Mounts can be issued from the network. This message is displayed in response to the **status** operand of the **modify** command.

**System Action:** Network File System processing continues.

---

**GFSA908I(<PROCNAME>) HFS PROCESSING  
DISABLED.**

**Explanation:** HFS processing is suspended. This message is displayed in response to the status operand of the **modify** command.

**System Action:** The NFS processing continues.

---

**GFSA909E(<PROCNAME>) UNMOUNT COMMAND  
FAILED: MOUNT POINT STILL IN USE**

**Explanation:** This message is in response to the **unmount** operand of the **modify** command. The unmount processing fails because the file system is still in use. The user can retry the **unmount** command later after there is no reference to file system.

**System Action:** The system ends the request. The system continues processing for the NFS.

---

**GFSA910I(<PROCNAME>) *text* ACTIVE = *d\_digits*.**

**Explanation:** This message is in response to the **list=mounts** operand of the **modify** command and shows the mounted file space (MVS path) and the number of active clients (*d\_digits*) mounting those files by that path. The number might not be accurate if a client has crashed without unmounting the files.

**System Action:** Network File System processing continues.

---

**GFSA911I(<PROCNAME>) *text1(text2)* ACTIVE =  
*d\_digits*.**

**Explanation:** This status message is in response to the **list=mounts** operand of the **modify** command and shows the mounted file space (MVS path) and the number of active clients (*d\_digits*) mounting those files by that path. The number might not be accurate if a client has crashed without unmounting the files.

**System Action:** Network File System processing continues.

---

**GFSA912I(<PROCNAME>) *text*.**

**Explanation:** The data set name appears in response to a **list=dsnames** operand of the **modify** command and shows a currently active data set.

**System Action:** Network File System processing continues.

---

**GFSA913I(<PROCNAME>) *text1(text2)*.**

**Explanation:** The PDS member appears in response to the **list=dsnames** operand of the **modify** command and shows a currently active data set.

**System Action:** Network File System processing continues.

---

**GFSA914I(<PROCNAME>) *text* DEALLOCATED.**

**Explanation:** The data set name *text* appears in response to a **release=dsname(member)** operand of the **modify** command after successful deallocation.

**System Action:** Network File System processing continues.

---

**GFSA915I(<PROCNAME>) *text* NOT ALLOCATED.**

**Explanation:** The data set name *text* appears in response to the **release=dsname(member)** operand of the **modify** command if the data set or member specified to be released was not found.

**System Action:** Network File System processing continues.

---

**GFSA916I(<PROCNAME>) *text* UNMOUNTED.**

**Explanation:** The data set name *text* appears in response to an **unmount** (data set or member) operand of the **modify** command after a successful unmount.

**System Action:** Network File System processing continues.

---

**GFSA917I(<PROCNAME>) *text* NOT MOUNTED.**

**Explanation:** The data set name *text* appears in response to the **unmount** (data set or member) operand of the **modify** command if the data set or member specified to be unmounted was not found in the current mount list.

**System Action:** Network File System processing continues.

---

**GFSA918I(<PROCNAME>) *text* IS NOT A VALID  
DATA SET NAME.**

**Explanation:** The data set name specified on either the **release** or **unmount** operand of the **modify** command is not a valid MVS data set name.

**System Action:** Network File System processing continues.

**Operator Response:** Specify the command again with a valid data set name.

---

**GFSA919I(<PROCNAME>) *text* IS NOT A VALID MEMBER NAME.**

---

**Explanation:** The member name specified on either the **release** or **unmount** operand of the **modify** command is not a valid MVS member name.

**System Action:** Network File System processing continues.

**Operator Response:** Specify the command again with a valid member name.

---

**GFSA920I(<PROCNAME>) NO ACTIVE MOUNT POINTS.**

---

**Explanation:** This message is in response to the **list=mounts** operand of the **modify** command and shows that there are no clients connected to the Network File System.

**System Action:** Network File System processing continues.

---

**GFSA921I(<PROCNAME>) NO ACTIVE DATA SETS.**

---

**Explanation:** This message is in response to the **list=dsnames** operand of the **modify** command and shows that there are no clients actively accessing data sets.

**System Action:** Network File System processing continues.

---

**GFSA922I(<PROCNAME>) VERIFY: (*text*) IS NOT A VSAM DATA SET.**

---

**Explanation:** This message is in response to the **verify** operand of the **modify** command and shows that the data set name specified is not a virtual storage access method (VSAM) data set.

In the message text:

*text*            The data set name.

**System Action:** The system ends the request. The system continues processing for the Network File System.

---

**GFSA923I(<PROCNAME>) VERIFY SUCCESSFUL FOR (*text*).**

---

**Explanation:** This message is in response to the **verify** operand of the **modify** command. It shows that the verify was successful for the VSAM data set.

In the message text:

*text*            The data set name.

**System Action:** The system continues processing for the Network File System.

---

**GFSA924I(<PROCNAME>) VERIFY FAILED WITH RC = *d\_digits* FOR (*text*).**

---

**Explanation:** This message is in response to the **verify** operand of the **modify** command. It shows that the verify failed with a return code for the data set. Message GFSA847I follows this message in the log data set.

In the message text:

*d\_digits*        The return code.

*text*            The data set name.

**System Action:** The system ends the request. The system continues processing for the Network File System.

---

**GFSA925I(<PROCNAME>) ERROR WAS DETECTED IN THE EXPORTS FILE. EXPORT LIST NOT REBUILT.**

---

**Explanation:** This is the reply from the **exportfs** operand of the **modify** command, indicating that one or more errors were detected in the exports data set; for example, the exports data set cannot be opened.

**System Action:** Network File System processing continues. The existing exports list is not changed.

**Operator Response:** Notify the system programmer.

**System Programmer Response:** Review previous console error messages for detailed information as to the specific error in the exports data set.

---

**GFSA926I(<PROCNAME>) EXPORT LIST HAS BEEN REBUILT SUCCESSFULLY.**

---

**Explanation:** This is the reply from the **exportfs** operand of the **modify** command, indicating that the command has completed normally.

**System Action:** Network File System processing continues with the new exports data set in effect.

**System Programmer Response:** Network File System processing continues with the new exports data set in effect.

---

**GFSA927I(<PROCNAME>) MODIFY EXPORTS COMMAND IGNORED - THE EXPORTS FILE IS NOT BEING USED FOR SECURITY CHECKING.**

---

**Explanation:** This is the reply from the **exportfs** operand of the **modify** command, indicating that the command was ignored because the site attribute for security requested that no exports file checking be done.

**System Action:** Network File System processing

continues with the security options unchanged.

---

**GFSA930I(<PROCNAME>) LOG DATA SET IS SWITCHED FROM *text1* TO *text2*.**

**Explanation:** A “no space” or an I/O error condition is detected when writing to the log data set. Network File System logging is now switched to the other log data set. *text1* is the DD associated with the switched-from log data set. *text2* is the DD associated with the switched-to log data set.

**System Action:** Network File System processing continues.

**Operator Response:** If requested by the installation, back up the switched-from log data set at this point. The switched-from data set is reused when the switched-to log data set is also filled.

**System Programmer Response:** Consider allocating larger NFS server log data sets for future Network File System usage. Note that the last data buffer is lost when the log is switched.

---

**GFSA931I(<PROCNAME>) NETWORK FILE SYSTEM SERVER LOGGING IS TERMINATED.**

**Explanation:** The Network File System logging is ended. This can be caused by a “no space” condition of the log data set.

**System Action:** Network File System processing continues.

**System Programmer Response:** Allocate a larger log data set for future Network File System usage.

---

**GFSA932I(<PROCNAME>) LOG DATA SET *text* IS FLUSHED.**

**Explanation:** The data buffer of the active log data set is flushed to disk. If the log is written to standard error, *text* is “STDERR”. Otherwise *text* is the associated DD of the active log data set.

**System Action:** Network File System processing continues.

---

**GFSA933I(<PROCNAME>) LOG DATA SET IS RE-INITIALIZED.**

**Explanation:** The log data set is re-initialized.

**System Action:** Network File System processing continues.

---

**GFSA934I(<PROCNAME>) NFSLOG1 OR NFSLOG2 DD STATEMENTS NOT DEFINED.**

**Explanation:** The “NFSLOG1” or the “NFSLOG2” DD statements are not coded in the Network File System start-up cataloged procedure.

**System Action:** The server stops.

**System Programmer Response:** Code the NFSLOG1 and NFSLOG2 DD statements and allocate the associated log data sets.

---

**GFSA935I(<PROCNAME>) SWITCHED-TO LOG IN ERROR. NETWORK FILE SYSTEM SERVER LOGGING CONTINUED ON DD:*text*.**

**Explanation:** An operator issued a **switchlog** command but an open error is detected in new log data set. Network File System logging is continued in the original log data set. *text* is the DD associated with the original log data set.

**System Action:** NFS processing continues.

**Operator Response:** Fix the non-active log data set.

---

**GFSA936I(<PROCNAME>) NETWORK FILE SYSTEM SERVER LOG *text* SET TO FORCELOG.**

**Explanation:** The Network File System server log was closed to force all log data to disk immediately.

**System Action:** The Network File System continues.

---

**GFSA949I COMMAND *text* NOT VALID.**

**Explanation:** The operator has entered the **operator** command or operand *text* which is not valid.

**System Action:** Network File System processing continues.

**Operator Response:** Enter the operand of the **modify** command again with the correct syntax.

---

**GFSA950I Unknown flag '*-character*'**

**Explanation:** The character *character* specified on the **mvslogin**, **mvslogout**, or **showattr** command is not a valid option. A usage message might follow this message.

**User Response:** See *z/OS Network File System User's Guide* for a description of the valid options used with the command.

---

**GFSA951I *text* : can't find name for uid *d\_digits*.**

**Explanation:** There was an error reading information for UID *d\_digits* from the *etc/passwd* file.

**User Response:** Correct the *etc/passwd* file and try the command again.

---

**GFSA952I    Retyped password does not match**

**Explanation:** The password entered when message GFSA975I was displayed does not match the password entered when message GFSA974A was displayed.

**User Response:** Start the **mvlogin** command sequence again.

---

**GFSA953I    Password change required by host.**

**Explanation:** The multiple virtual system (MVS) password for the user ID passed to the host has expired. A new password is required. Message GFSA974I follows this message.

---

**GFSA954I    Host *text1* returned error *d\_digits: text2***

**Explanation:** An error was detected during **mvlogin** processing. Host *text1* returned error code *d\_digits* and message *text2* to the client.

**User Response:** The password or user ID might be incorrect. Start the **mvlogin** command sequence again and use the correct password or user ID.

---

**GFSA955I    *text* logged in ok.**

**Explanation:** The MVS user ID *text* was logged in without any errors.

---

**GFSA956I    usage: *text* [-pn][-g group][-a account] hostname [**mv\_username**]**

**Explanation:** Usage information for the **text** command.

**User Response:** Enter the command using the correct syntax.

---

**GFSA957I    Host *text1* returned error *d\_digits: text2***

**Explanation:** An error was detected during **mvlogout** processing. Host *text1* returned error code *d\_digits* and message *text2* to the client.

**User Response:** Notify your MVS system programmer.

---

**GFSA958I    uid *text* logged out ok.**

**Explanation:** The MVS user ID *text* was logged out successfully.

---

**GFSA959I    usage: *text* hostname**

**Explanation:** Usage information for the *text* command.

**User Response:** Enter the command using the correct syntax.

---

---

**GFSA960I    *text1*: host "*text2*" unknown.**

**Explanation:** The host *text2*, specified on command *text1*, is not known to the network.

**User Response:** Correct the host name specified and try the command again.

---

**GFSA961I    *text1*: *text2***

**Explanation:** Command *text1* received an error when trying to create a client transport handle using the **clntudp\_create** TCP/IP remote procedure call. *text2* is the message produced by the **clnt\_pcreateerror** TCP/IP procedure. This message is issued if:

- The host name is unknown
- The host is not operational
- The Network File System on the named host is not operational

**User Response:** The user response is:

- Correct the host name specified and try the command again,
  - Make sure the specified host is operational and try the command again, or
  - Make sure the NFS server on the named host is operational and try the command again.
- 

**GFSA964I    *text*: Error: cannot determine server.**

**Explanation:** The *text* command found the mount path, but the server name was not returned by the local operating system service that keeps mount point information.

**User Response:** Correct the mount point table and try the command again.

---

**GFSA965I    *text1*: Error: *text2* mounted from server *text3*, not *text4*.**

**Explanation:** The wrong host name was specified for the *text1* command. *text2* is mounted from server *text3* instead of server *text4*.

**User Response:** Specify the command again with the correct host name.

---

**GFSA966I    *text*: Error: unknown return from usage routine.**

**Explanation:** The usage routine used for the **text** command returned an unknown error code.

**User Response:** Contact your programming support personnel.

---

---

**GFSA967I** Host Error: *text*.

**Explanation:** The host returned an error and message *text*. This might be due to:

- A porting failure occurred for the **showattr** command.
- The Network File System is not compatible with the **showattr** command on the client.

**User Response:** Contact your programming support personnel.

---

**GFSA968I** Error: Drive *text* not mounted.

**Explanation:** The drive *text* was not mounted. The **showattr** command cannot show the attributes for this drive.

**User Response:** Mount the drive and reissue the command.

---

**GFSA969I** Error: Can't open *text* for read.

**Explanation:** The file *text* could not be opened to read the mount path.

**User Response:** Correct the file and reissue the command.

---

**GFSA970I** Error: Directory *text* not mounted.

**Explanation:** The directory *text* was not mounted. The **showattr** command cannot show the attributes for this directory.

**User Response:** Mount the directory and reissue the command.

---

**GFSA971I** Error: filesystem *text* is local.

**Explanation:** The file system *text* is not a Network File System file system. The **showattr** command is for Network File System file systems only.

**User Response:** Reissue the command for a Network File System file system.

---

**GFSA972I** usage: *text1* [-t] hostname [*text2*]

**Explanation:** Usage information for the *text1* command. *text2* is the operating system dependent mount point format.

---

**GFSA973A** Enter MVS password:

**Explanation:** The Network File System requires a password for the user.

**User Response:** If an MVS user ID was specified on the **mvslogin** command, enter the password for that MVS user ID. If no MVS\_user ID was specified, the name from *etc/passwd* for the UID that issued the **mvslogin** command was passed to the Network File System. Enter the MVS password for this user.

---

**GFSA974A** Enter new MVS password:

**Explanation:** The MVS password for the user ID passed to Network File System has expired.

**User Response:** Enter a new MVS password.

---

**GFSA975A** Retype new MVS password:

**Explanation:** MVS requires the new password to be entered twice for verification.

**User Response:** Enter the new MVS password again.

---

**GFSA976I** *text1*: *text2*

**Explanation:** Command *text1* received an error when trying to create a client transport handle using the *clnt\_call* TCP/IP remote procedure call. *text2* is the message produced by the *clnt\_perror* TCP/IP procedure.

**User Response:** Contact your system administrator.

---

**GFSA977I** *text*:

**Explanation:** Command *text* received an error when trying to create a client transport handle using the *clnt\_call* TCP/IP remote procedure call (RPC). This message is followed by the message produced by the *clnt\_perror* TCP/IP procedure.

**User Response:** Contact your system administrator.

---

**GFSA978I** *text* logged in ok. Mismatch in uid/gid:  
OpenEdition uid is *digit\_1*, gid is  
*digit\_2*, client uid is *digit\_3*, gid is *digit\_4*

**Explanation:** The OpenEdition UID/GID does not match the client machine UID/GID. The authentication is successful and the message is for informational use only.

---

**GFSA991E(<PROCNAME>) MESSAGE FORMAT  
FROM USER EXIT ROUTINE(S) IS  
INCORRECT. USER *text* EXIT  
ROUTINE(S) HAS(HAVE) ENDED.**

**Explanation:** The *test* exit routine returned an incorrect length message. *text* can be either LOGON AND FILE SECURITY or FILE SECURITY.

**System Action:** The NFS server processing continues without the user exit routine(s).

**Operator Response:** Record MVS operator console message and notify the NFS system programmer.

**System Programmer Response:** Correct user exit routine(s), relink the user exit routine(s), and restart the NFS server.

---

**GFSA996E(<PROCNAME>) INSTALLATION  
DEFAULT TRANSLATION TABLE  
CANNOT BE INITIALIZED.**

**Explanation:** The Network File System ends because one of these conditions happens during Network File System startup:

- The NFSXLAT DD statement is not coded in the Network File System start-up catalog procedure, and the **xlat** processing attribute is specified.
- Translation table data set defined in the NFSXLAT DD statement is not a PDS or PDSE.
- translation table specified in the translation table data set cannot be found.
- Translation table contained in the translation table data set is in an incorrect format.

**System Action:** Network File System processing stops.

**System Programmer Response:** Do the following before restarting the Network File System:

- Code the NFSXLAT DD statement in the Network File System start-up catalog procedure correctly.
- Allocate the translation table data set defined in the NFSXLAT DD statement.
- Make sure the translation table in the translation table data set exists.
- Make sure the translation table specified in the translation table data set is in the correct format.

unsuccessful. *text* is the member name that contains the translation table. The cause might be:

- The NFSXLAT DD statement is not coded in the Network File System start-up catalog procedure.
- The NFSXLAT DD statement is not coded correctly in the Network File System start-up catalog procedure.
- Translation table data set defined in the NFSXLAT DD statement is not a PDS or PDSE.
- translation table specified in the translation table data set cannot be found.
- The format of the translation table contained in the translation table data set is not valid.

**System Action:** Network File System processing stops.

**User Response:** This error occurs on mount operation. Contact your system programmer for the correct member name for the translation table.

---

**GFSA998I TRANSLATION TABLE *text* IS LOADED.**

**Explanation:** Network File System loaded the translation table, *text*, successfully.

**System Action:** Network File System processing continues.

---

**GFSA997I ERROR IN READING TRANSLATION  
TABLE, *text*.**

**Explanation:** The Network File System tried to read the translation table, *text*, during the mount but was

---

## Client messages

This is a listing of the messages generated by the NFS client. A message, explanation, and recommended action is supplied where applicable. Data is substituted for any part of a message shown here in *italics*.

Messages that appear on the MVS operator's console for the NFS client will be in the following example format: **GFSC700I OS/390 VERSION 1 NETWORK FILE SYSTEM CLIENT(HDZ11TC) STARTED**

Table 31 shows the NFS client MVS operators console message format:

*Table 31. NFS client MVS operators console message format*

GFSC	Component identifier for the NFS client.
700	A unique message number.

Table 31. NFS client MVS operators console message format (continued)

<b>I</b>	Message type:
<b>A</b>	Action; the user must perform a specific action.
<b>E</b>	Eventual action; the user must perform an action when time is available.
<b>I</b>	Informational; no user action is required.
<b>OS/390 VERSION 1 NETWORK FILE SYSTEM CLIENT(HDZ11TC) STARTED</b>	Message text.

Messages appear in the NFS client log data set in the same format as the following example: **12:34:18 GFSC100E (E) CCXDR 11 XDR\_DISP: RPC REQUEST (7F638E30) FAILED, RETURN VALUE -1 RETURN CODE 00000467 REASON CODE 5 (TIMED OUT)**

Table 32 shows the NFS client client log data set message format:

Table 32. NFS client log data set message format

<b>12:34:18</b>	The time stamp (hours:minutes:seconds).
<b>GFSC</b>	Component identifier for the NFS client.
<b>100</b>	A unique message number.
<b>E</b>	Message type:
<b>A</b>	Action; the user must perform a specific action.
<b>E</b>	Eventual action; the user must perform an action when time is available.
<b>I</b>	Informational; no user action is required.
<b>(E)</b>	The message level: E (error), W (warning), or I (informational).
<b>CCXDR 11</b>	'CCXDR' is the last 5 characters of CSECT name. '11' is the message sequence within the function.
<b>XDR_DISP</b>	The first 8 characters of the function name.
<b>RPC REQUEST (7F638E30) FAILED, RETURN VALUE -1 RETURN CODE 00000467 REASON CODE 5 (TIMED OUT)</b>	Message text.

The messages are listed in numerical order (the time stamp, message level, and programming support information are not shown).

**Notes:**

1. *h\_digits* denotes hexadecimal digits, and *d\_digits* denotes decimal digits. *text* represents variable text (such as a data set name).
2. Messages GFSC098I and GFSC099I are intended for IBM support personnel when they are performing diagnosis.

Table 33 shows the common variables in the message text:

*Table 33. Common variables*

<b>Variable</b>	<b>Meaning</b>
retv	decimal return value
retc	decimal return code
rsnc	decimal reason code
returncd	8-digit hexadecimal return code
reasoncd	8-digit hexadecimal reason code
h_digit	8-digit hexadecimal address
d_digit	decimal digits
text	Place holder for long text of different lengths

Table 34 can be used for initial translation of the reason code, *reasoncd*, information presented in messages related to parsing errors.

*Table 34. Parsing error (when reason code is 6E01xxxx)*

<b>Last 4 hex digits of <i>reasoncd</i></b>	<b>Reason</b>
Fxxx	Unknown keyword
11yy	Host name
12yy	Path name
13yy	Keyword <b>acdirmax</b>
14yy	Keyword <b>acdirmin</b>
15yy	Keyword <b>acregmax</b>
16yy	Keyword <b>acregmin</b>
17yy	Keyword <b>cln_ccsid</b>
18yy	Keyword <b>srv_ccsid</b>
19yy	Keyword <b>hard</b>
1Ayy	Keyword <b>soft</b>
1Byy	Keyword <b>retrans</b>
1Cyy	Keyword <b>timeo</b>
1Dyy	Keyword <b>wsiz</b>
1Eyy	Keyword <b>rsiz</b>
1Fyy	Keyword <b>retry</b>
2Ayy	Keyword <b>vers</b>
2Byy	Keyword <b>proto</b>
21yy	Keyword <b>Biod</b>
22yy	Keyword <b>Bufhigh</b>
23yy	Keyword <b>DelayWrite</b>
24yy	Keyword <b>ReadAhead</b>
25yy	Keyword <b>AttrCaching</b>
26yy	Keyword <b>DataCaching</b>
27yy	Keyword <b>DynamicSizeAdj</b>
28yy	Keyword <b>delim</b>
29yy	Keyword <b>xlat</b>



Table 34. Parsing error (when reason code is 6E01xxxx) (continued)

Last 4 hex digits of <i>reasoncd</i>	Reason
<b>Notes:</b>	
<b>xxx</b>	Offset to the beginning of the <b>mount</b> parameter to the bad keyword
<b>yy</b>	See Table 35 for more details

Table 35 can be used for further translation of the reason code, *reasoncd*, information presented in messages related to parsing errors.

Table 35. Parsing error (when reason code is from 6E0111yy to 6E0129yy)

Last 2 hex digits of <i>reasoncd</i>	Reason
01	Null host name or null path name
02	Blank detected
03	Incorrect member name in the path name
04	Missing double quote
05	No member name found
06	Missing left parenthesis
07	Incorrect number
08	Number is larger than 2G
09	Incorrect multiplier, must be K, M, or G
0A	Missing right parenthesis
0B	The specified number is not within the allowable range
0C	Incorrect keyword parameter value
0D	Mutually exclusive keyword/option
0E	Keyword is not allowed in the mount option
0F	Keyword is not allowed in the installation parameter

Table 36 contains the rest of the reason code, *reasoncd*, information presented in messages.

Table 36. Parsing error (when reason code is 6Exxyyyy')

xx of <i>reasoncd</i>	Module	yyyy of <i>reasoncd</i>	Description
02	TCP/IP common error	0001	<i>clntudp_create( )</i> failed
02	TCP/IP common error	0002	Server NFS port not 2049
02	TCP/IP common error	0003	<i>authunix_create( )</i> failed
02	TCP/IP common error	0004	<i>clnt_control( )</i> timeout failed
02	TCP/IP common error	0005	<i>clnt_control( )</i> total timeout failed
02	TCP/IP common error	0006	<i>clnttcp_create( )</i> failed
02	TCP/IP common error	0007	<i>clntudp_bufcreate( )</i> failed
02	TCP/IP common error	1num	<i>clnt_call( )</i> timeout (:retc=0467h, ETIMEDOUT)
02	TCP/IP common error	1num	<i>clnt_call( )</i> EINTR (:retc=0078h, EINTR)

Table 36. Parsing error (when reason code is 6Exyyyy') (continued)

<i>xx of reasoncd</i>	<b>Module</b>	<i>yyyy of reasoncd</i>	<b>Description</b>
03	TCP/IP error	yyyy	TCP error - <i>clnt_control( )</i> failed
03	TCP/IP error	yyyy	TCP error - <i>authunix_create( )</i> failed
03	TCP/IP error	yyyy	TCP error - <i>clnt_call( )</i> failed
04	TCP/IP error	yyyy	Authentication error <i>authunix_create( )</i> failed
05	NFS protocol error	0001	00001, not owner (:retc=0088h, EPERM)
05	NFS protocol error	0002	00002, no such file/directory (:retc=0081h, ENOENT)
05	NFS protocol error	0005	00005, I/O error (:retc=007Ah, EIO)
05	NFS protocol error	0006	00006, no such device or address (:retc=008Ah, ENXIO)
05	NFS protocol error	000D	00013, permission denied (:retc=006Fh, EACCESS)
05	NFS protocol error	0011	00017, file/dir exists (:retc=0075h, EEXIST)
05	NFS protocol error	0012	00018, cross-device link (:retc=0090h, EXDEV)
05	NFS protocol error	0013	00019, no such device (:retc=0080h, ENODEV)
05	NFS protocol error	0014	00020, not a directory (:retc=0087h, ENOTDIR)
05	NFS protocol error	0015	00021, is a directory (:retc=0078h, EISDIR)
05	NFS protocol error	0016	00022, incorrect arguments (:retc=0079h, EIVAL)
05	NFS protocol error	001B	00027, file too large (:retc=0077h, EFBIG)
05	NFS protocol error	001C	00028, no space left on device (:retc=0085h, ENOSPC)
05	NFS protocol error	001E	00030, Read-only file system (:retc=008Dh, EROFS)
05	NFS protocol error	001F	00031, too many links (:retc=007Dh, EMLINK)
05	NFS protocol error	003F	00063, file name too long (:retc=007Eh, ENAMETOOLONG)
05	NFS protocol error	0042	00066, directory not empty (:retc=0088h, ENOTEMPTY)
05	NFS protocol error	0045	00069, disk quota exceeded (:retc=046Dh, EDQUOT)
05	NFS protocol error	0046	00070, stale file handle (:retc=046Eh, ESTALE)
05	NFS protocol error	0047	00071, too many levels of remote (:retc=046Fh, EREMOTE)
05	NFS protocol error	2711	10001, bad file descriptor (:retc=0071h, EBADF)

Table 36. Parsing error (when reason code is 6Exyyyy') (continued)

<i>xx of reasoncd</i>	<b>Module</b>	<i>yyyy of reasoncd</i>	<b>Description</b>
05	NFS protocol error	2712	10002, not sync (:retc=0071h, EBADF)
05	NFS protocol error	2713	10003, bad cookie (:retc=0076h, EFAULT)
05	NFS protocol error	2714	10004, operation not support (:retc=0086h, ENOSYS)
05	NFS protocol error	2715	10005, buffer too small (:retc=0462h, ENOBUFS)
05	NFS protocol error	2716	10006, server fault (:retc=007Ah, EIO)
05	NFS protocol error	2717	10007, bad type (:retc=008Ah, ENXIO)
05	NFS protocol error	2718	10008, jukebox (:retc=0070h, EAGAIN)
0D	UNIX System Services (USS)	0001	Abend, no SDWA (:retc=009Dh, EMVSERR)
0D	UNIX System Services (USS)	0002	Unknown Abend (:retc=009Dh, EMVSERR)
0E	UNIX System Services (USS)	yyyy	System Abend Code (:retc=009Dh, EMVSERR)
0F	UNIX System Services (USS)	yyyy	User Abend Code (:retc=009Dh, EMVSERR)
10	GFSCMAIN	<i>Inum</i>	main
11	GFSCVNAC	<i>Inum</i>	open, close, create, access, fsync, trunc, inactive
12	GFSCVNAT	<i>Inum</i>	lookup, getattr, setattr, pathconf
13	GFSCVNDR	<i>Inum</i>	mkdir, rmdir, readdir
14	GFSCVNLK	<i>Inum</i>	link, symlink, readlink
15	GFSCVNRM	<i>Inum</i>	rename, remove, recovery
16	GFSCVNRW	<i>Inum</i>	read, write
18	GFSCVMNT	<i>Inum</i>	mount, umount, statfs, pfscctl, sync, recovery
19	GFSCCMNT	<i>Inum</i>	common Rnode attributes management
1A	GFSCREQB	<i>Inum</i>	reqblock/credential allocate/free
21	GFSCBIOD	<i>Inum</i>	BioD - Buffer input/output (I/O) daemon
22	GFSCMNTD	<i>Inum</i>	MntD - Mount daemon
23	GFSCBIOA	<i>Inum</i>	Common buffer management
24	GFSCBIOB	<i>Inum</i>	Buffer management
25	GFSCBIOC	<i>Inum</i>	Buffer input/output (I/O)
26	GFSCXDR2	<i>Inum</i>	XDR routines for NFS version 2 protocol
27	GFSCXDR3	<i>Inum</i>	XDR routines for NFS version 3 protocol
28	GFSC1RPC	<i>Inum</i>	Thread exit
29	GFSC0RPC	<i>Inum</i>	Thread and daemon starter
31	GFSCLOCK	<i>Inum</i>	Lock management

---

**GFSC100E**    **RPC REQUEST (*h\_digit*) FAILED,  
RETURN VALUE -1 RETURN CODE  
*returncd* REASON CODE *rsnc* (*text*)**

**Explanation:** Remote procedure call (RPC) request failed. (*h\_digit*) is the request block address. Reason code *rsnc*, is the return code returned from TCP/IP. *text* is the failure reason. For the explanation of return code *returncd*; see *z/OS UNIX System Services Messages and Codes*.

**System Action:** Client continues processing.

**User Response:** See the return code *returncd* in the OpenEdition message manual, and reason code *rsnc* in the TCP/IP message manual.

---

**GFSC101E**    **NETWORK FILE SYSTEM SERVER  
REQUEST FAILED (*h\_digit*),  
OPENEDITION RETURN CODE *returncd*  
NETWORK FILE SYSTEM SERVER  
RETURN CODE *retc* (*text*)**

**Explanation:** The NFS server failed the request from the client. (*h\_digit*) is the request block address. Return code *retc* is returned from NFS server. *text* is the failure reason for the request. For the explanation of return code *returncd*; see *z/OS UNIX System Services Messages and Codes*.

**System Action:** Client continues processing.

**User Response:** See the return code *returncd* in OpenEdition message manual, and the return code *retc* in the Network File System Protocol Specification, RFC 1094.

---

**GFSC102E**    **RPC REQUEST (*h\_digit*) FAILED,  
RETURN VALUE -1 RETURN CODE  
*returncd* REASON CODE *rsnc***

**Explanation:** Remote procedure call (RPC) request failed. (*h\_digit*) is the request block address. Reason code *rsnc*, is the return code returned from TCP/IP. The explanation of return code *returncd* is described in the OpenEdition message manual.

**System Action:** Client continues processing.

**User Response:** See the return code, *returncd*, in OpenEdition message manual, and the reason code, *rsnc*, in the TCP/IP message manual.

---

**GFSC103E**    **NETWORK FILE SYSTEM SERVER  
REQUEST FAILED (*h\_digit*),  
OPENEDITION RETURN CODE *returncd*  
NETWORK FILE SYSTEM SERVER  
RETURN CODE *retc***

**Explanation:** The NFS server failed the request from the client. (*h\_digit*) is the request block address. Return code *retc* is returned from the NFS server. For the

explanation of return code *returncd*; see the OpenEdition message manual.

**System Action:** Client continues processing.

**User Response:** See the return code *returncd* in OpenEdition message manual, and the return code *retc* in the Network File System Protocol Specification, RFC 1094.

---

**GFSC105E**    **READ FAILED, RETURN VALUE -1  
RETURN CODE *returncd* REASON  
CODE *reasoncd*.**

**Explanation:** While reading a block of data from a remote file, an error *returncd* was detected.

**System Action:** The read operation ends. NFS client processing continues.

**User Response:** See *z/OS UNIX System Services Messages and Codes* for a description of the return code *returncd* and the reason code *reasoncd* to determine corrective action.

---

**GFSC106E**    **WRITE FAILED, RETURN VALUE -1  
RETURN CODE *returncd* REASON  
CODE *reasoncd*.**

**Explanation:** While writing a block of data to a remote file, an error *returncd* was detected.

**System Action:** The write operation ends. NFS client processing continues. The remote file may not be complete.

**User Response:** See *z/OS UNIX System Services Messages and Codes* for a description of the return code *returncd* and the reason code *reasoncd* to determine corrective action.

---

**GFSC107E**    **FLUSH FAILED, RETURN VALUE -1  
RETURN CODE *returncd* REASON  
CODE *reasoncd*.**

**Explanation:** While flushing cached data to a remote file during close processing, an error *returncd* was detected.

**System Action:** The write operation ends. NFS client processing continues. The remote file may not be complete.

**User Response:** See *z/OS UNIX System Services Messages and Codes* for a description of the return code *returncd* and the reason code *reasoncd* to determine corrective action.

---

**GFSC110E** *text* FAILED, RETURN VALUE -1  
RETURN CODE *returncd* REASON  
CODE *reasoncd*

**Explanation:** The NFS client has detected an error in the function, *text*.

**System Action:** The request has ended. NFS client processing continues.

**User Response:** See *z/OS UNIX System Services Messages and Codes* for a description of the return code *returncd* and the reason code *reasoncd* to determine corrective action.

**System Programmer Response:** Collect the detail trace log from client and from server if any.

---

**GFSC200E** VFS\_MOUNT FAILED, RETURN VALUE  
-1 RETURN CODE *returncd* REASON  
CODE *reasoncd*.

**Explanation:** The **mount** command failed because of error *returncd*.

**System Action:** The **mount** command ended abnormally. NFS client processing continues.

**User Response:** See *z/OS UNIX System Services Messages and Codes* for a description of the return code *returncd* and the reason code *reasoncd* to determine the corrective action, and reissue **mount** command.

---

**GFSC201E** NETWORK FILE SYSTEM CLIENT  
DOES NOT SUPPORT SYNCHRONOUS  
MOUNT REQUEST.

**Explanation:** NFS client only supports asynchronous mount.

**System Action:** The **mount** command ended with an error. NFS client processing continues.

**User Response:** Reissue **mount** command with the asynchronous option.

---

**GFSC202E** A FILE SYSTEM WITH THE SAME  
NAME IS ALREADY MOUNTED.

**Explanation:** Cannot mount on an existing mount point.

**System Action:** The **mount** command ended with an error. No mount point was established. NFS client processing continues.

**User Response:** Reissue **mount** command with a different mount point.

---

**GFSC203E** PARSING MOUNT OPTION FAILED,  
RETURN VALUE -1 RETURN CODE  
*returncd* REASON CODE *reasoncd*  
OPTION=*'text'*.

**Explanation:** The mount option *text* was incorrectly specified.

**System Action:** The **mount** command ended with an error. No mount point is established. NFS client processing continues.

**User Response:** See *z/OS UNIX System Services Messages and Codes* for a description of the return code *returncd*, and see Table 34 on page 120 for further details of the reason code *reasoncd*. Correct the mount option, and reissue the **mount** command .

---

**GFSC204E** VFS\_UMOUNT FAILED, RETURN  
VALUE -1 RETURN CODE *returncd*  
REASON CODE *reasoncd*.

**Explanation:** The **umount** or **unmount** command failed.

**System Action:** The **umount** or **unmount** command ended with an error. The mount point may still exist. NFS client processing continues.

**User Response:** See *z/OS UNIX System Services Messages and Codes* for a description of the return code *returncd* and the reason code *reasoncd* to determine the corrective action. Correct it and reissue **umount** again.

---

**GFSC205E** VFS\_STATFS FAILED, RETURN VALUE  
-1 RETURN CODE *returncd* REASON  
CODE *reasoncd*.

**Explanation:** The VFS\_STATFS operation failed. While trying to get the status of a remote file system, an error *returncd* was detected.

**System Action:** The VFS\_STATFS ended with an error. NFS client processing continues.

**User Response:** See *z/OS UNIX System Services Messages and Codes* for a description of the return code *returncd* and the reason code *reasoncd* to determine the corrective action.

---

**GFSC206E** VFS\_SYNC FAILED, RETURN VALUE -1  
RETURN CODE *returncd* REASON  
CODE *reasoncd*.

**Explanation:** The VFS\_SYNC operation failed. While flushing cached data of remote files, an error *returncd* was detected.

**System Action:** The VFS\_SYNC ended with an error. The remote files may not be complete. NFS client processing continues.

**User Response:** See *z/OS UNIX System Services*

*Messages and Codes* for a description of the return code *returncd* and the reason code *reasoncd* to determine the corrective action.

---

**GFSC207E VFS\_RECOVER FAILED, RETURN VALUE -1 RETURN CODE *returncd* REASON CODE *reasoncd*.**

**Explanation:** The VFS\_RECOVER operation failed. While trying to recover from a previous abend, an error *returncd* was detected.

**System Action:** The VFS\_RECOVER ended with an error. NFS client processing continues.

**User Response:** See *z/OS UNIX System Services Messages and Codes* for a description of the return code *returncd* and the reason code *reasoncd* to determine the corrective action. If the error occurs and the program is not in error, look at the messages in the client log data sets for more information. Search problem reporting data bases for a fix for the problem. If no fix exists, contact the IBM Support Center. Provide all the printed output and copies of output data sets related to the problem.

---

**GFSC208E VFS\_PFSCTL FAILED, RETURN VALUE -1 RETURN CODE *returncd* REASON CODE *reasoncd*.**

**Explanation:** The VFS\_PFSCTL operation failed.

**System Action:** The VFS\_PFSCTL ended with an error. NFS client processing continues.

**User Response:** See *z/OS UNIX System Services Messages and Codes* for a description of the return code *returncd* and the reason code *reasoncd* to determine the corrective action. If the error occurs and the program is not in error, look at the messages in the client log data sets for more information. Search problem reporting data bases for a fix for the problem. If no fix exists, contact the IBM Support Center. Provide all the printed output and copies of output data sets related to the problem.

---

**GFSC209E THE IP ADDRESS OF REMOTE HOST NAME, *hostname*, COULD NOT BE RESOLVED.**

**Explanation:** The VFS\_MOUNT operation failed. The mount processing failed when it tried to resolve the remote host name *hostname* to a dotted IP address.

**System Action:** The **mount** command ended with an error. NFS client processing continues.

**User Response:** Correct the remote host name or use the dotted IP address of the remote host, and reissue the **mount** command.

---

**GFSC210E NFS SERVER *<hostname>* DOES NOT SUPPORT NFS VERSION 3 PROTOCOL WHILE 'VERS(3)' WAS SPECIFIED**

**Explanation:** The VFS\_MOUNT operation failed. The mount processing failed because the server does not support NFS version 3 protocol, which the user requested with the **mount** parameter *vers(3)*.

**System Action:** The **mount** command ended with an error. NFS client processing continues.

**User Response:** Verify that the server actually does not support NFS version 3 protocol. Remove *vers(3)* from the **mount** parameter, and reissue the **mount** command.

---

**GFSC211E NFS SERVER *<hostname>* DOES NOT SUPPORT 'AUTH\_UNIX' AUTHENTICATION.**

**Explanation:** The VFS\_MOUNT operation failed. The **mount** processing failed because the server does not do UNIX authentication. (AUTH\_UNIX). OS/390 does not support other authentication such as Kerberos.

**System Action:** The **mount** command ended with an error.

**User Response:** Verify that the server actually does not support AUTH\_UNIX authentication. Notify the server system administrator.

---

**GFSC212E MOUNT FAILED BECAUSE OF CONVERSION SERVICE CONNECTION FAILURE CCSID *ccsid* RETC= *retc* RSNC= *rsnc***

**Explanation:** CCSIDs specified in the **mount** command are not supported by conversion service.

**System Action:** The **mount** command ended abnormally. NFS client continues processing.

**User Response:** See *z/OS UNIX System Services Messages and Codes* for more information about the return code *returncd* and the reason code *reasoncd* to determine corrective action

**System Programmer Response:** Check availability of specified CCSIDs.

---

**GFSC213E REQUEST (*requestid*) : THE NFS SERVER *hostname* DOES NOT SUPPORT THE SPECIFIED 'VERS' AND/OR 'PROTO'.**

**Explanation:** The **mount** request with the *requestid* failed because the user-specified *vers* or *proto* was not supported by the NFS server on the *hostname*.

**System Action:** The **mount** command ended with an error. NFS client continues processing.

**User Response:** Either let the NFS client choose the

compatible *vers* and *proto*, or determine the NFS server capabilities (by *orpcinfo*) and re-issue the **mount** command with the proper *vers* or *proto*.

---

**GFSC214E REQUEST (*requestid*) : THE NFS SERVER *hostname* DOES NOT HAVE NFS REGISTERED ON PORT 2049**

**Explanation:** The request with the *requestid* failed because the NFS server on the *hostname* did not register or use port 2049.

**System Action:** The operation ended with an error. NFS client continues processing.

**User Response:** Use *orpcinfo* to verify the server and correct the server.

---

**GFSC300E MISSING LEFT PARENTHESIS IN *text* KEYWORD.**

**Explanation:** Specified keyword, *text*, is missing a left parenthesis.

**System Action:** The NFS client processing stops if the error occurs in the NFS client installation parameter. The **mount** command failed if the error is in the **mount** parameter.

**Operator Response:** Record MVS operator console message and notify the MVS system programmer.

**System Programmer Response:** If the error is in the **mount** parameter, correct the parameter, and reissue **mount** command. If the error occurs in the NFS client installation parameter, correct the parameter, stop OpenEdition, and restart OpenEdition.

---

**GFSC301E PARSE FAILED ON NUMERIC FIELD FOR *text* KEYWORD.**

**Explanation:** Specified keyword *text* contains alphabetic data in numeric field.

**System Action:** NFS client processing stops if the error is in the NFS client installation parameter. The **mount** command failed if the error is in the **mount** parameter.

**Operator Response:** Record MVS operator console message and notify the MVS system programmer.

**System Programmer Response:** If the error is in the **mount** parameter, correct the parameter, and reissue **mount** command. If the error occurs in the NFS client installation parameter, correct the parameter, stop OpenEdition, and restart OpenEdition.

---

**GFSC302E MISSING RIGHT PARENTHESIS IN *text* KEYWORD.**

**Explanation:** Specified keyword *text* is missing a right parenthesis.

**System Action:** The NFS client processing stops if

the error is in the NFS client installation parameter. The **mount** command failed if the error is in the **mount** parameter.

**Operator Response:** Record MVS operator console message and notify the MVS system programmer.

**System Programmer Response:** If the error is in the **mount** parameter, correct the parameter, and reissue **mount** command. If the error occurs in the NFS client installation parameter, correct the parameter, stop OpenEdition, and restart OpenEdition.

---

**GFSC303E THE SPECIFIED VALUE *d\_digit1* IS NOT IN THE RANGE OF *d\_digit2* TO *d\_digit3* FOR *text* KEYWORD.**

**Explanation:** The value, *d\_digit1*, specified in the keyword, *text*, must be between the minimum value, *d\_digit2*, and the maximum value, *d\_digit3*.

**System Action:** The NFS client processing stops if the error is in the NFS client installation parameter. The **mount** command failed if the error is in the **mount** parameter.

**Operator Response:** Record MVS operator console message and notify the MVS system programmer.

**System Programmer Response:** If the error is in the **mount** parameter, correct the parameter, and reissue **mount** command. If the error occurs in the NFS client installation parameter, correct the parameter, stop OpenEdition, and restart OpenEdition.

---

**GFSC304E PARSE FAILED ON ALPHABETIC FIELD FOR *text* KEYWORD.**

**Explanation:** Specified keyword, *text*, contains numeric data for alphabetic field.

**System Action:** The NFS client processing stops if the error is in the NFS client installation parameter. The **mount** command failed if the error is in the **mount** parameter.

**Operator Response:** Record MVS operator console message and notify the MVS system programmer.

**System Programmer Response:** If the error is in the **mount** parameter, correct the parameter, and reissue **mount** command. If the error occurs in the NFS client installation parameter, correct the parameter, stop OpenEdition, and restart OpenEdition.

---

**GFSC305E INCORRECT OPTION *text1* SPECIFIED FOR *text2* KEYWORD, VALID OPTION IS Y OR N.**

**Explanation:** An incorrect option, *text1*, was specified for the keyword, *text2*.

**System Action:** The NFS client processing stops if the error is in the NFS client installation parameter. The

**mount** command failed if the error is in the **mount** parameter.

**Operator Response:** Record MVS operator console message and notify the MVS system programmer.

**System Programmer Response:** If the error is in the **mount** parameter, correct the option for the keyword *text2* and reissue **mount** command. If the error occurs in the NFS client installation parameter, correct the option for the keyword, *text2*, stop OpenEdition, and restart OpenEdition.

---

**GFSC307E** *text* IS AN INCORRECT KEYWORD FOR MOUNT PARAMETER.

**Explanation:** *text* can only be specified as an installation parameter.

**System Action:** The **mount** command failed.

**System Programmer Response:** Correct the **mount** parameter keyword *text*.

---

**GFSC308E** REQUEST (*requestid*) : XLAT(Y) CANNOT BE SPECIFIED AS A MOUNT PARAMETER WHEN TAG OPTION IS ALSO SPECIFIED.

**Explanation:** The **mount** request with the *requestid* code failed because the user specified both *xlat(Y)* and *tag* options together. This is not allowed.

**System Action:** The **mount** command ended with an error. NFS client continues processing.

**User Response:** Either specify *xlat(Y)* to have the NFS client do text translation based on the *cln\_ccsid* and *srv\_ccsid* values, or specify *tag* option with the proper Coded Character Set Identifier (CCSID) to have the translation done by LFS based on the CCSID in the *tag* option.

---

**GFSC309E** UNKNOWN KEYWORD ENCOUNTERED AROUND POSITION *d\_digit*.

**Explanation:** The keyword specified in position *d\_digit* is not a valid keyword.

**System Action:** The NFS client processing stops if the error is in the NFS client installation parameter. The **mount** command failed if the error is in the **mount** parameter.

**Operator Response:** Record MVS operator console message and notify the MVS system programmer.

**System Programmer Response:** If the error is in the **mount** parameter, correct the keyword, and reissue the **mount** command. If the error occurs in the NFS client installation parameter, correct the keyword, stop OpenEdition, and restart OpenEdition.

---

**GFSC310I** READAHEAD AND DELAYWRITE OPTIONS WILL BE IGNORED AS DATACACHING IS OFF.

**Explanation:** The keywords *ReadAhead* and *DelayWrite* will be ignored because **DataCaching** has been set OFF.

**System Action:** NFS client continues processing.

**Operator Response:** Record MVS operator console message and notify the MVS system programmer.

**System Programmer Response:** Check parameters to make sure that **DataCaching** should be OFF.

---

**GFSC311I** CLN\_CCSID AND SRV\_CCSID WILL BE IGNORED AS XLAT OPTION IS OFF.

**Explanation:** The keywords *cln\_ccsid* and *srv\_ccsid* will be ignored because **xlat** has been set off.

**System Action:** NFS client continues processing.

**Operator Response:** Record MVS operator console message and notify the MVS system programmer.

**System Programmer Response:** Check parameters to make sure that **xlat** should be off.

---

**GFSC312I** ACREGMIN, ACREGMAX, ACDIRMIN, AND ACDIRMAX OPTIONS WILL BE IGNORED AS ATTRCACHING IS OFF.

**Explanation:** The keywords *acregmin*, *acregmax*, *acdirmin*, and *acdirmax* will be ignored because **AttrCaching** has been set OFF.

**System Action:** NFS client continues processing.

**Operator Response:** Record MVS operator console message and notify the MVS system programmer.

**System Programmer Response:** Check parameters to make sure that **AttrCaching** should be OFF.

---

**GFSC313I** RETRANS OPTION WILL BE IGNORED AS HARD OPTION IS ON.

**Explanation:** The keyword *retrans* will be ignored because *hard* has been set ON.

**System Action:** NFS client continues processing.

**Operator Response:** Record MVS operator console message and notify the MVS system programmer.

**System Programmer Response:** Check parameters to make sure that *hard* should be ON.

---

**GFSC315E** ERROR ENCOUNTERED WHILE PARSING MOUNT PATH, REASON CODE *reasoncd*.

**Explanation:** The specified mount path is not correct.

**System Action:** The **mount** command failed.



**Operator Response:** Record MVS operator console message and notify the MVS system programmer.

**System Programmer Response:** See *z/OS UNIX System Services Messages and Codes* for a description of the reason code *reasoncd*. Correct the mount path, and reissue mount command.

---

**GFSC317E ERROR ENCOUNTERED WHILE  
PARSING HOSTNAME, REASON CODE  
*reasoncd*.**

**Explanation:** The specified hostname is not correct.

**System Action:** The mount command failed.

**Operator Response:** Record MVS operator console message and notify the MVS system programmer.

**System Programmer Response:** See *z/OS UNIX System Services Messages and Codes* for a description of the reason code *reasoncd*. Correct the mount path, and reissue mount command.

---

**GFSC318E READ FAILED FOR NETWORK FILE  
SYSTEM CLIENT MOUNT  
PARAMETERS.**

**Explanation:** An error occurred while the NFS client was processing the **mount** parameters. This message follows other messages, GFSC3xxE, that describe the error in greater detail.

**System Action:** The mount command failed.

**System Programmer Response:** Correct the **mount** parameter options, and reissue **mount** command.

---

**GFSC319E *text* IS AN INCORRECT KEYWORD FOR  
NETWORK FILE SYSTEM CLIENT  
INSTALLATION PARAMETER.**

**Explanation:** *text* can only be specified as a mount parameter.

**System Action:** NFS client processing stops.

**Operator Response:** Record MVS operator console message and notify the MVS system programmer.

**System Programmer Response:** Correct the NFS client installation parameter, stop OpenEdition, and restart OpenEdition.

---

**GFSC320E INCORRECT OPTION *text* SPECIFIED  
FOR DELIM KEYWORD, VALID OPTION  
IS BINARY, CR, CRLF, CRNL, LF, LFCR,  
OR NL.**

**Explanation:** An incorrect option *text*, has been specified for the keyword *delim*.

**System Action:** NFS client processing stops if the error is in the NFS client installation parameter. The

**mount** command failed if the error is in the **mount** parameter.

**Operator Response:** Record MVS operator console message and notify the MVS system programmer.

**System Programmer Response:** If the error is in the mount parameter, correct the option for the keyword *delim*, and reissue mount command. If the error occurs in the NFS client installation parameter, correct the option for the keyword *delim*, stop OpenEdition, and restart OpenEdition.

---

**GFSC500I CLIENT LOG DATA SET, *text*,  
FLUSHED.**

**Explanation:** The data buffer of the active client log data set *text*, was flushed to disk. *text* is the associated data set name of the active client log data set.

**System Action:** NFS client processing continues.

---

**GFSC501I CLIENT LOG DATA SET *text*  
RE-INITIALIZED.**

**Explanation:** The error log data set is re-initialized. *text* is the associated data set name of the active log data set.

**System Action:** NFS client processing continues.

---

**GFSC502E CANNOT OPEN CLIENT LOG DATA  
SET, *text1*, *text2*.**

**Explanation:** NFS client failed to open client log data set. *text1* is the DD associated with the client log data set which cannot be opened. *text2* is the failure reason of the *C* function, *fopen*.

**System Action:** NFS client processing stops.

**Operator Response:** Record MVS operator console message and notify the MVS system programmer.

**System Programmer Response:** Fix client log data set and reply to the OpenEdition message to restart the NFS client.

---

**GFSC503E CLIENT LOGGING ENDED.**

**Explanation:** NFS client failed to manipulate the client log data set. See previous operator console message for the failure reason.

**System Action:** NFS client processing continues.

**Operator Response:** Record MVS operator console message and notify the MVS system programmer.

**System Programmer Response:** Fix client log data set, stop the NFS client, and reply to the OpenEdition message to restart the NFS client.

---

**GFSC504I    CLIENT LOG DATA SET SWITCHED TO  
text.**

**Explanation:** A "no space" or an I/O error condition was detected while writing to the client log data set. NFS client logging switched to the other log data set, *text*.

**System Action:** NFS client processing continues.

---

**GFSC505E    MISSING DD STATEMENT OR  
INCORRECT DATA SET  
ORGANIZATION FOR LOG DATA SET.**

**Explanation:** The error log data set has incorrect data set organization, or missing DD statement.

**System Action:** NFS client processing stops if the error occurred during initialization time. NFS client processing continues with client logging ended if the error occurred after initialization time.

**Operator Response:** Record MVS operator console message and notify the MVS system programmer.

**System Programmer Response:** Correct error log data set DD statement or data set organization, stop the NFS client, and reply to the OpenEdition message to restart the NFS client.

---

**GFSC506E    text1 FAILED FOR text2, text3.**

**Explanation:** NFS client failed to manipulate the client log data set, *text2*. *text3* is the failure reason for the C function, *text1*.

**System Action:** NFS client processing continues.

**Operator Response:** Record MVS operator console message and notify the MVS system programmer.

**System Programmer Response:** Fix client log data set, stop the NFS client, and reply to the OpenEdition message to restart the NFS client.

---

**GFSC700I    OS/390 version NETWORK FILE  
SYSTEM CLIENT fmid STARTED**

**Explanation:** The NFS client is initialized and ready to process NFS requests.

In the message text:

*version*        The version number, for example:  
VERSION 1.

*fmid*            The NFS client FMID started, for example:  
(HDZ11TC).

**System Action:** NFS client continues processing.

---

**GFSC701I    NETWORK FILE SYSTEM CLIENT  
SHUTDOWN IN PROGRESS.**

**Explanation:** NFS client shutdown processing has started.

**System Action:** NFS client shutdown processing continues.

---

**GFSC702I    NETWORK FILE SYSTEM CLIENT  
SHUTDOWN COMPLETE.**

**Explanation:** NFS client has completed shutdown processing.

**System Action:** NFS and its associated subtasks have ended.

---

**GFSC703E    NETWORK FILE SYSTEM CLIENT  
INITIALIZATION FAILED: NETWORK  
FILE SYSTEM CLIENT IS ALREADY  
STARTED.**

**Explanation:** Only one NFS client can be started on an MVS system.

**System Action:** This NFS client ends.

**Operator Response:** Record MVS operator console message and notify the MVS system programmer.

**System Programmer Response:** A prior NFS client session has not ended. UNIX end processing should have ended the NFS client colony address space. Collect installation parameters, dumps, NFS client log data sets, and contact IBM programming support personnel. Use the force command to end the NFS client colony address space and then restart UNIX.

---

**GFSC704E    NETWORK FILE SYSTEM CLIENT  
INITIALIZATION FAILED: DOWN LEVEL  
SECURITY PRODUCT.**

**Explanation:** Resource Access Control Facility (RACF) MVS security product is down level.

**System Action:** The NFS client ends.

**Operator Response:** Record MVS operator console message and notify the MVS system programmer.

**System Programmer Response:** Check to determine the required RACF level.

---

**GFSC705E    NETWORK FILE SYSTEM CLIENT  
INITIALIZATION FAILED: MAIN  
STORAGE IS UNAVAILABLE.**

**Explanation:** The NFS client was not able to allocate the necessary storage. The cause might be the value specified on the *bufhigh* attribute is too large or the REGION size is too small.

**System Action:** The NFS client ends.

**Operator Response:** Record MVS operator console message and notify the MVS system programmer.

**System Programmer Response:** The requested memory is not available during the NFS client initialization. Do one or both of the following before restarting the NFS client:

- Increase the REGION size for the client procedure.
- Decrease the value specified for the bufhhigh attribute of the FILESYSTYPE parameter in the BPXPRMxx Parmlib member.

---

**GFSC707E NETWORK FILE SYSTEM CLIENT  
INITIALIZATION FAILED: INCORRECT  
PARAMETER IN INSTALLATION  
PARAMETERS.**

**Explanation:** The NFS client has detected an error in the installation parameters.

**System Action:** The NFS client ends.

**Operator Response:** Record MVS operator console message and notify the MVS system programmer.

**System Programmer Response:** Check previous console messages prefixed with GFSC. Correct the parameter, stop OpenEdition, and restart OpenEdition.

---

**GFSC708E NETWORK FILE SYSTEM CLIENT  
INITIALIZATION FAILED: CONVERSION  
SERVICE IS NOT INSTALLED OR NOT  
AVAILABLE. RETURN CODE *retc*,  
REASON CODE *rsnc*.**

**Explanation:** Both Unicode and Character Data Representation Architecture (CDRA) initialization requests failed.

**System Action:** The NFS client startup ends.

**Operator Response:** Record MVS operator console message and notify the MVS system programmer.

**System Programmer Response:** See *z/OS MVS System Messages, Vol 2 (ASB-EWX)* for a description of the return code *retc* and the reason code *rsnc* to determine the corrective action.

---

**GFSC709E NETWORK FILE SYSTEM CLIENT  
INITIALIZATION FAILED: NOT  
STARTED IN A STANDALONE COLONY  
ADDRESS SPACE.**

**Explanation:** The NFS client has detected an error during initialization processing. the NFS client was started by some other means, other than UNIX kernel.

**System Action:** The NFS client ends.

**Operator Response:** Record MVS operator console message and notify the MVS system programmer.

**System Programmer Response:** The NFS client must be initialized by UNIX kernel.

---

**GFSC710E NETWORK FILE SYSTEM CLIENT  
INITIALIZATION FAILED: ESTAE  
INITIALIZATION FAILED.**

**Explanation:** The NFS client has detected an error during the initialization processing.

**System Action:** The NFS client ends.

**Operator Response:** Collect any dumps, NFS client log data sets, and notify the system programmer.

**System Programmer Response:** Collect installation parameters, dumps, NFS client log data sets, and Contact IBM programming support personnel.

---

**GFSC711E NETWORK FILE SYSTEM CLIENT  
INITIALIZATION FAILED: OPENEDITION  
KERNEL SERVICE FAILED.**

**Explanation:** The NFS client has detected an error during the second phase of initialization processing.

**System Action:** The NFS client ends.

**Operator Response:** Collect any dumps, NFS client log data sets, and notify the system programmer.

**System Programmer Response:** Collect installation parameters, dumps, NFS client log data sets, and contact IBM programming support personnel.

---

**GFSC712E NETWORK FILE SYSTEM CLIENT  
INITIALIZATION FAILED: SOCKET  
CALL GETHOSTNAME FAILED,  
RETURN CODE *returncd*.**

**Explanation:** The NFS client has detected an error (*returncd*) during the initialization processing. This error may be caused by OpenEdition and TCP/IP connection failure.

**System Action:** The NFS client ends.

**Operator Response:** Collect any dumps, NFS client log data sets, and notify the system programmer.

**System Programmer Response:** Collect installation parameters, dumps, NFS client log data sets, and Contact IBM programming support personnel.

---

**GFSC713E NETWORK FILE SYSTEM CLIENT  
LOGGING IS TERMINATED.**

**Explanation:** The NFS client has detected an error. The Client Log daemon has ended.

**System Action:** The NFS client ends.

**Operator Response:** Collect any dumps, NFS client log data sets, and notify the system programmer.

**System Programmer Response:** Collect installation parameters, dumps, NFS client log data sets, and Contact IBM programming support personnel.

---

**GFSC714E NETWORK FILE SYSTEM CLIENT SVC  
DUMP REQUEST FAILED, RETURN  
CODE *returncd* REASON CODE  
*reasoncd*.**

**Explanation:** A request to write an MVS SVC dump failed. See the description of the **sdump** macro in *z/OS MVS Programming: Authorized Assembler Services Reference LLA-SDU* for the meaning of the return code *returncd* and reason code *reasoncd*.

**System Action:** The NFS client ends.

**Operator Response:** Collect the MVS console log, NFS client log data sets, and notify the system programmer.

**System Programmer Response:** Collect installation parameters, dumps, NFS client log data sets, and Contact IBM programming support personnel.

---

**GFSC715E NETWORK FILE SYSTEM CLIENT  
ESTAE EXIT UNABLE TO COMPLETE,  
ABEND CODE *d\_digit*.**

**Explanation:** The NFS client recovery exit has detected a recursive abend. The *d\_digit* is the last abend code encountered by a secondary instance of the ESTAE exit routine.

**System Action:** The ESTAE processing stops. The NFS client ends.

**Operator Response:** Collect any dumps, the MVS console log, NFS client log data sets, and notify the system programmer.

**System Programmer Response:** Collect installation parameters, dumps, the MVS console log, NFS client log data sets, and Contact IBM programming support personnel.

---

**GFSC716I NETWORK FILE SYSTEM CLIENT  
DAEMON TASK RESTARTED.**

**Explanation:** The NFS client detected a stopped asynchronous daemon task. The NFS client has restarted the stopped asynchronous daemon task.

**System Action:** The NFS client daemon has restarted. NFS client processing continues.

---

**GFSC721E UNABLE TO SETUP ERROR  
RECOVERY (ESTAE), RETURN CODE  
*returncd*.**

**Explanation:** NFS client daemon or thread failed to setup error recovery. See *z/OS MVS Programming: Authorized Assembler Services Reference ENF-IXG* for information the **estae** macro return code, *returncd*.

**System Action:** If the error occurred in the daemon, the NFS client has initiated shutdown processing. If the error occurred in the thread, the associated operation

ends and NFS client continues processing.

**Operator Response:** Collect any dumps, the MVS console log, NFS client log data sets, and notify the system programmer.

**System Programmer Response:** If the error occurs and the program is not in error, look at the messages in the client log data sets for more information. Search problem reporting data bases for a fix for the problem. If no fix exists, contact the IBM Support Center. Provide all the printed output and copies of output data sets related to the problem.

---

**GFSC722E A SOCKET COULD NOT BE CREATED,  
RETURN VALUE -1 RETURN CODE  
*returncd* REASON CODE *reasoncd*.**

**Explanation:** NFS client Daemon or thread processing failed to create a socket for network communication.

**System Action:** If the error occurred in the daemon, the NFS client has initiated shutdown processing. If the error occurred in the thread, the associated operation ends and the NFS client continues processing.

**Operator Response:** Collect any dumps, the MVS console log, NFS client log data sets, and notify the system programmer.

**System Programmer Response:** See *z/OS UNIX System Services Messages and Codes* for a description of the return code *returncd* and the reason code *reasoncd* to determine the corrective action. If the error occurs and the program is not in error, look at the messages in the client log data sets for more information. Search problem reporting data bases for a fix for the problem. If no fix exists, contact the IBM Support Center. Provide all the printed output and copies of output data sets related to the problem.

---

**GFSC723E NETWORK FILE SYSTEM CLIENT  
ABEND @ *module+offset*.**

**Explanation:** NFS client encountered a programming error. SVC dump was issued to capture the diagnostic information.

**System Action:** NFS client has initiated shutdown processing.

**System Programmer Response:** Look at the messages in the client log data sets for more information. Search problem reporting data bases for a fix for the problem. If no fix exists, contact the IBM Support Center. Provide all the printed output and copies of output data sets related to the problem.

---

**GFSC724E UNABLE TO BIND A RESERVED PORT  
TO SOCKET *socketnum*.**

**Explanation:** The mount daemon or umount thread failed to obtain a reserved port for socket, *socketnum*, for network communication.

**System Action:** If the error occurred in the daemon, the NFS client has initiated shutdown processing. If the error occurred in the thread, the associated operation ends and the NFS client continues processing.

**System Programmer Response:** If the error occurs and the program is not in error, look at the messages in the client log data sets for more information. Search problem reporting data bases for a fix for the problem. If no fix exists, contact the IBM Support Center. Provide all the printed output and copies of output data sets related to the problem.

---

**GFSC725E SOCKET, *socketnum*, COULD NOT BE CLOSED, RETURN VALUE -1 RETURN CODE *returncd* REASON CODE *reasoncd*.**

**Explanation:** While closing a socket *socketnum*, an error *returncd* was detected.

**System Action:** NFS client continues processing.

**System Programmer Response:** See *z/OS UNIX System Services Messages and Codes* for a description of the return code *returncd* and the reason code *reasoncd* to determine the corrective action. If the error occurs and the program is not in error, look at the messages in the client log data sets for more information. Search problem reporting data bases for a fix for the problem. If no fix exists, contact the IBM Support Center. Provide all the printed output and copies of output data sets related to the problem.

---

**GFSC726E REQUEST (*requestid*) : THE PORTMAPPER OF THE SERVER *hostname* DOES NOT RESPOND TO PMAP\_GETMAPS**

**Explanation:** The **mount** request with the *requestid* code failed because the user specified *proto(tcp)* and the *hostname* did not publish its registered remote program port for remote procedure call (RPC).

**System Action:** The **mount** command ended with an error. NFS client continues processing.

**User Response:** Verify that the server does not respond to *orpcinfo*. Re-issue the **mount** command without *proto(tcp)*.

---

**GFSC727W REQUEST (*requestid*) : THE PORTMAPPER OF THE SERVER *hostname* DOES NOT RESPOND TO PMAP\_GETMAPS. ATTEMPTING UDP RPC WITH NFS VERSION *d\_digit* PROTOCOL.**

**Explanation:** The NFS client warns the user who mounts to the *hostname* that the host may not have its **Portmapper** running; the NFS client uses user datagram protocol (UDP) remote procedure call (RPC). A user datagram protocol (UDP) remote procedure call

(RPC) is attempted with NFS protocol.

**System Action:** The **mount** command may or may not succeed. NFS client continues processing.

**User Response:** Verify that the server does not respond to the *orpcinfo* code. If the **mount** command subsequently fails, then it is likely that the host does not have the NFS server.

---

**GFSC728E REQUEST (*requestid*) : UNABLE TO CONNECT SOCKET=*socketno* PORT=*portno1* WITH THE SERVER *hostname* PORT=*portno2***

**Explanation:** The NFS client uses TCP/IP to connect the socket *socketno* with port *portno1* to the server *hostname* with port *portno2*. However, the connect system call failed.

**System Action:** The operation ended with an error. The NFS client continues processing.

**User Response:** Verify that the NFS server on *hostname* is available and running. Verify that the TCP/IP subsystem on MVS is available and running. If the **mount** command subsequently fails, then it is very likely that the host does not have the NFS server.

---

**GFSC729W OS/390 NFS CLIENT IS UNABLE TO RESERVE *d\_digit* SOCKETS, CURRENT MAXFILEPROC IS *d\_digit***

**Explanation:** The maximum number of socket descriptors that a process can have open concurrently has been exceeded. Increase the MAXFILEPROC parameter value in the BPXPRMxx member to bypass this problem. See *z/OS UNIX System Services Planning* for more information about specifying the MAXFILEPROC value.

**System Action:** The operation ended with an error. The NFS client continues processing.

**System Programmer Response:** Increase the MAXFILEPROC parameter value in the BPXPRMxx member to bypass the problem.

---

**GFSC840I usage: *text* [-a] [-d] [-e] [*host*]**

**Explanation:** This is the usage for the **showmount** command. *text* is the command as entered by the user. The valid options are as follows:

- a Display all mounts in the format  
Hostname:Directory from **host** NFS server
- d Display only directory names of all mounts from  
**host** NFS server
- e Display the list of exported directories from **host**  
NFS server

---

**GFSC841E Unknown host *text***

**Explanation:** The user entered incorrect host address information, *text*.

**System Action:** Command stops processing.

**User Response:** Correct syntax and re-issue the command.

---

**GFSC842E Cannot resolve local host name**

**Explanation:** Local host name is not found.

**System Action:** Command stops processing.

**User Response:** Contact your system administrator to check TCP/IP configuration.

---

**GFSC843E Unknown flag '*-character*'**

**Explanation:** An incorrect option, '*-character*', is specified.

**System Action:** Command stops processing.

**User Response:** Correct syntax and re-issue the command.

---

**GFSC845I usage: *text* input output**

**Explanation:** This is the usage for the **os22mvs** and **mvs2os2** commands. *text* is the command as entered by the user. Table 37 shows the valid parameters:

*Table 37. GFSC845I valid parameters*

<i>input</i>	Absolute path name of the input file to be converted.
<i>output</i>	Absolute path name of the output file.

---

**GFSC846E Cannot open input file, *text1: text2***

**Explanation:** Cannot open input file, *text1*. *text1* is the input path name as entered by the user. *text2* is the failure information returned when attempting to open the input file.

**System Action:** Command stops processing.

**User Response:** Check input file, *text1*.

---

**GFSC847E Cannot open output file, *text1: text2***

**Explanation:** Cannot open output file, *text1*. *text1* is the output path name as entered by the user. *text2* is the failure information returned when attempting to open the output file.

**System Action:** Command stops processing.

**User Response:** Check output file, *text1*.

---

**GFSC848E Cannot read input file, *text1: text2***

**Explanation:** Cannot read input file, *text1*. *text1* is the input path name as entered by the user. *text2* is the failure information returned when attempting to read the input file.

**System Action:** Command stops processing.

**User Response:** Check input file, *text1*.

---

**GFSC849E Cannot write output file, *text1: text2***

**Explanation:** Cannot write output file, *text1*. *text1* is the output path name as entered by the user. *text2* is the failure information returned when attempting to write the output file.

**System Action:** Command stops processing.

**User Response:** Check output file, *text1*.

---

**GFSC850E Input path name cannot be equal to output path name.**

**Explanation:** Input path name cannot be equal to output path name.

**System Action:** Command stops processing.

**User Response:** Correct syntax and re-issue the command.

---

**GFSC854I usage: *text* [-crnzm <mount point> ]**

**Explanation:** This is the usage for the **nfsstat** command. *text* is the command as entered by the user. The valid parameters are as follows:

- c Display both NFS and remote procedure call (RPC) statistics about the NFS client
- n Display NFS statistics about the NFS client
- r Display remote procedure call (RPC) statistics about the NFS client
- z Initializes statistics to zero. This is for use by root user only and can be combined with any of the above options. Zero particular set of statistics after printing them.
- m Display the name of each NFS mounted file system
- m *mount point*  
Display information of NFS mounted file system on the specified mount point

---

**GFSC855E Must be a root user to issue '*character*' flag**

**Explanation:** The option, '*character*', can only be issued with the root authority.

**System Action:** Command stops processing.

**User Response:** Contact your system administrator to issue this command.

---

**GFSC856E** Network File System Client command, *text*, failed, return value -1 return code *reasoncd* reason code *reasoncd*

**Explanation:** The command, *text*, failed.

**System Action:** Command stops processing.

**User Response:** See *z/OS UNIX System Services Messages and Codes* for a description of the return

code, *reasoncd*. See Table 34 on page 120 for more information about the reason code, *reasoncd*.

---

**GFSC858E** Directory *text* not mounted.

**Explanation:** The directory, *text*, was not mounted.

**System Action:** Command stops processing.

**User Response:** Issue `nfsstat -m` to view the list of active mount points. If the mount point does not exist, contact system administrator to mount the directory.

---

## Messages from the client platform (AIX)

This section provides a list of messages from the client operating system, in response to NFS reply results (messages without message numbers). These messages are platform dependent.

**Note:** The message text is from an AIX RS/6000.

This section contains messages from the client's operating system.

---

### Cross device link

**Explanation:**

1. An attempt has been made to rename a member of a PDS or PDSE, but the target file is not a member of the same PDS or PDSE.
2. An attempt has been made to rename a non-PDS or PDSE file, but the target file is a member of a PDS or PDSE.

**User Response:** Try a copy and remove instead of a rename.

---

### Directory Not Empty

**Explanation:** An attempt has been made to remove a PDS or PDSE that has members.

**User Response:** Delete members before trying to remove a PDS or PDSE.

---

### File exists

**Explanation:** An attempt has been made to rename a PDS or PDSE, but the target file already exists.

**User Response:** Delete the target file before renaming a PDS or PDSE. This is not required for a regular file or for a PDS or PDSE member.

---

### File Name Too Long

**Explanation:** The name is not a valid MVS file or member name.

**User Response:** File names must follow the MVS naming conventions. See *z/OS DFSMS: Using Data Sets* for MVS file naming conventions.

---

### Invalid

**Explanation:**

1. The specified parameters were incorrect.
2. The creation of a VSAM data set failed.

**User Response:** Respecify the correct parameters or contact the system programmer to determine the VSAM data set failure on the NFS server.

---

### I/O Error (with possible system programmer response)

**Explanation:**

1. Unexpected error from Catalog Management.
2. Dynamic Allocation failed during action other than a read or write.
3. A file could not be opened during an action other than a read or write.
4. An error occurred while reading a partitioned data set (PDS) or PDSE directory.
5. No space is available in task input/output table (TIOT).
6. TIOT resource is unavailable.
7. Unable to release enough resources.
8. Insufficient units are available.
9. The server could not get enough memory to perform this function.

**System Programmer Response:** For (9), stop the server and change the region field in the job control language (JCL) before restarting, or modify parameters in the attributes file. For the other possible explanations, perform the appropriate action.

---

### I/O Error (with possible user response)

#### Explanation:

1. An attempt has been made to nest PDSs or PDSEs.
2. Maximum number of file allocations is exceeded.
3. The file is being written in text mode, the new write request offset is determined to fall within the end-of-line (EOL) sequence (lf, cr, crlf, lfcr) of a previous line, and the new data does not contain the correct EOL characters.
4. The file is being written in text mode, with a non-zero EOL (lf, cr, lfcr, crlf). The number of bytes of data in the written line is larger than the maximum record size of the file.
5. The file is being written in text mode, with fixed records, with a non-zero EOL, **blankstrip** not set (no padding blanks on write), and the number of bytes of data in the written line is less than or greater than the record size of the file.
6. The file is being written in text mode, with fixed records, **blankstrip** set, and the line of written data contains trailing blanks as part of the data.
7. When a workstation file containing a 0 length line is written to MVS as recfm(u) in text mode, a write error occurs.
8. An MVS Access Method Services alias name was specified in a remove (**rm** or **rmdir**) or rename (**mv**) request.
9. An "s" was specified in the recfm attribute for a PDS or PDSE.
10. If you try to append data to a member of a PDS or PDSE, an I/O error occurs.
11. An incorrect attribute was specified in the command.

**User Response:** Perform the appropriate action.

---

### Is a directory

**Explanation:** A non-directory operation has been tried on a PDS or PDSE.

**User Response:** Use directory operations on the file.

---

### Network File System server *name* not responding still trying

**Explanation:** Long delays between operations.

**User Response:** The user response is:

- The server might need extra time to service client requests. Wait until the message "Network File System server *name* ok" appears.
- Determine if the network traffic is heavy and overloaded. Try to isolate the path where the client workstation communicates with the server machine.

- Determine if the client workstation transmits the same requests over and over. Commands such as **nfsstat -c** on AIX or UNIX platforms show the number of client retransmissions (**retrans**) as well as the number of **badcalls** and **badxid**. When the number of **badcalls** and **badxid** are high, the client machine usually has bad retransmissions. High retransmissions might be caused by an overloaded network or a slow server.
- If the network is overloaded, contact your network administrator.
- If the server is slow, determine if your client workstation tends to transmit requests out-of-sequence or incompletely, and you are performing I/O to a file in text mode. If the request is incomplete, the client may not send the remainder of the request until a later time. In this case, increase the server's **cachewindow** attribute. The **cachewindow** buffers store out-of-sequence and incomplete requests from clients. The general rule in setting the *n* value of **cachewindow(n)** is  $n = ((\text{num of BIOD} + 1) * (\text{client\_max\_IO\_buffer\_size} / \text{transfer\_size}))$  where,
  - BIOD is the number of blocked I/O daemons set by the client workstation. This value is usually set to defaults at the installation of the operating system or by your system administrator.
  - **client\_max\_IO\_buffer\_size** is the amount of I/O data that is requested by the client (for example, the client writes 8192 bytes of data to the remote file system). This value is determined by your application programs.
  - **transfer\_size** is the actual size of data being sent across the network (for example, the 8192 bytes of data may be broken down to 16 smaller packets of 512 bytes (16x512=8192)). This value is determined dynamically by your client workstation.
- If your client workstation tends to send duplicate transmissions for the same request too often, thus increasing the workload on the server, you might want to delay the client's retransmission rate and request's timeout. On the mount command, you can specify: **mount -o retrans=3,timeout=30 IO\_buffer\_size**, where:

**retrans** Is the number of retransmissions allowed before a timeout. Default is vendor-specific ranging from **3-5**.

**timeout** Is the timeout value in tenths of a second. **timeout=30** means 3 seconds. Default is vendor specific ranging from **5-11**.

If the reply is not received by the client within the timeout period, a minor timeout has occurred for this request. The timeout period is doubled, and the



request is sent again. The process is repeated until the retransmission count specified by the **retrans** is reached; if no reply has been received, major timeout has occurred.

---

**No such device****Explanation:**

1. The file resides on an off-line device.
2. The file has been migrated to another storage level. Whether it is being recalled depends on the **retrieve** attribute.

**User Response:** For (1), contact your MVS operator or MVS system programmer. For (2), try the request again later, if the **retrieve** attribute is enabled. If not, try the request again, with the **retrieve** attribute enabled.

---

**No space left on device****Explanation:**

1. The file has exceeded the space that was allocated to it.
2. The PDS or PDSE has exceeded the space that was allocated to it.
3. The PDS directory has exceeded the space that was allocated to it.

**User Response:**

1. For (1), save this file into a larger file and then rename it to the old name, if necessary.
2. For (2), create a larger PDS or PDSE and store this member there. Then copy the members of the old PDS or PDSE to the new PDS or PDSE. Rename the new PDS or PDSE if necessary.
3. For (3), Create a new PDS with a larger directory (use the **dir** attribute). Store this member in the new PDS. Then copy the members from the old PDS to the new PDS and rename them, if necessary.

---

**No such file or directory**

**Explanation:** A **locate** command failed for this file. The file is not cataloged, or the MVS system operator might have unmounted the file before you issued the **umount** command.

**User Response:** Check your spelling. If it is correct, contact your system administrator.

---

**Not a directory**

**Explanation:** A directory operation has been tried on a file that is not a PDS or PDSE.

**User Response:** Use non-directory operations on the file.

---

**Not Owner****Explanation:**

1. File has not timed out yet.
2. File is open. Another client (maybe even the same client) has the file open for writing.
3. The client tried to change the mode in a **nfsattr** Network File System procedure call.

**User Response:** The user response is:

- For (1), follow the steps in waiting and retrying that are described under the "Permission denied" message.
- For (3), do not try to change the mode.

---

**Permission denied****Explanation:**

1. The file is not in use, but has not timed out yet (occurs most often when writing PDS or PDSE members without **writetimeout** seconds expiring between saving members).
2. The file is in use by an MVS user, or another client.
3. Dynamic allocation—Authorized function requested by an unauthorized user. The error codes from dynamic allocation are printed to the log data set.
4. Resource access control facility (RACF) is not active.
5. Dynamic allocation—Request denied by operator (dsname allocation). The error codes from dynamic allocation are printed to the log data set.
6. Dynamic allocation—Installation validation routine denied this request. The error codes from dynamic allocation are printed to the log data set.
7. You are not authorized for this request.
8. The MVS operator suspended mount request processing (only if this message appears following a mount attempt).
9. The file/prefix is not exported to your client (only if this message appears following a mount attempt).
10. IDCAMS failed during a rename or remove procedure. Usually this happens because the file is in use. The output from IDCAMS is printed to the log data set.
11. With OS/2, you might get a "SYS0055 Access denied" message if the **noretrieve** attribute is set and a **dir** command is done against a mounted file system containing migrated files.

**User Response:** For (1), To determine if this is the problem, check the timeout values (attrtimeout, readtimeout, writetimeout). Retry the request after the shortest timeout has expired. If the request still fails, retry the request after the next shortest timeout has expired. If it still fails, retry after the longest timeout has expired. If the request still fails, this is not the problem. For (2) and (10), try the request again later. For (3), (4),

(5), (6), and (9), notify your MVS system programmer. For (7), enter the **mvslogin** command again and retry the request. If the request still fails, notify your MVS system programmer. For (8), notify your MVS operator or MVS system programmer. For (11), specify the **retrieve** attribute on the **mount** command, or the MVS system administrator can make that the default.

---

### Read Only File System

**Explanation:** One of these Network File System procedures was tried on a read-only file system: link, write, rename, remove, mkdir, or create.

**User Response:** See the documentation on the exports data set to see how a file system is designated read-only (see *z/OS Network File System Customization and Operation*). The exports data set needs to be changed, or you are using it incorrectly.

---

### Stale NFS File Handle

**Explanation:** A file handle is used by the client and server sides of the Network File System to specify a particular file or prefix. A stale file handle occurs when the name is no longer valid, possibly due to one of the these conditions:

1. The file or prefix has been removed by the MVS operator.
2. The server has been stopped and brought back up. This affects files and members below mount points.

**User Response:** For (1), unmount and mount again. If your client maintains that the device is busy even though it is not, you might have to restart your client. For (2), enter the **mvslogin** command again and retry the request.

---

### Weak Authorization

**Explanation:** The authorization data in the remote procedure call (RPC) message was not valid. This is a client side error.

**User Response:** UNIX-style authorization is required.

## Appendix B. Return codes

Table 38 lists the externalized return codes that are defined by the NFS Version 2 protocol.

*Table 38. Externalized return codes defined by the NFS version 2 protocol*

Return Value	Return Code	Description
NFS_OK	0	Requests completed successfully and the results are valid.
NFSERR_PERM	1	Not owner. The caller does not have correct ownership to perform the requested operation.
NFSERR_NOENT	2	No such file or directory. The file or directory specified does not exist.
NFSERR_IO	5	A hard error occurred when the operation was in progress. For example, this could be a disk error.
NFSERR_NXIO	6	No such device or address.
NFSERR_ACCESS	13	Permission denied. The caller does not have the correct permission to perform the requested operation.
NFSERR_EXIST	17	File exists. The file specified already exists.
NFSERR_NODEV	19	No such device.
NFSERR_NOTDIR	20	Not a directory. The caller specified a non-directory in a directory operation.
NFSERR_ISDIR	21	Is a directory. The caller specified a directory in a non-directory operation.
NFSERR_EINVAL	22	An argument was passed to the z/OS NFS server was not valid.
NFSERR_FBIG	27	File too large. The operation caused a file to grow beyond the server's limit.
NFSERR_NOSPC	28	No space left on device. The operation caused the server's file system to reach its limit.
NFSERR_ROFS	30	Read-only file system. Write tried on a read-only file system.
NFSERR_NAMETOOLONG	63	File name too long. The file name in an operation was too long.
NFSERR_NOTEMPTY	66	Directory not empty. Tried to remove a directory that was not empty.
NFSERR_DQUOT	69	Disk quota exceeded. The client's disk quota on the server has been exceeded.
NFSERR_STALE	70	The file handle given in the arguments was not valid. That is, the file referred to by that file handle no longer exists, or access to it has been revoked.

Table 39 lists the externalized return codes that are defined by the NFS version 3 Protocol.

*Table 39. Externalized return codes defined by the NFS version 3 protocol*

Return Value	Return Code	Description
NFS_OK	0	Requests completed successfully and the results are valid.

Table 39. Externalized return codes defined by the NFS version 3 protocol (continued)

Return Value	Return Code	Description
NFS3ERR_PERM	1	Not owner. The caller does not have correct ownership to perform the requested operation.
NFS3ERR_NOENT	2	No such file or directory. The file or directory specified does not exist.
NFS3ERR_IO	5	A hard error occurred when the operation was in progress. For example, this could be a disk error.
NFS3ERR_NXIO	6	No such device or address.
NFS3ERR_ACCESS	13	Permission denied. The caller does not have the correct permission to perform the requested operation.
NFS3ERR_EXIST	17	File exists. The file specified already exists.
NFS3ERR_XDEV	18	Attempt to do an operation across the file system.
NFS3ERR_NODEV	19	No such device.
NFS3ERR_NOTDIR	20	Not a directory. The caller specified a non-directory in a directory operation.
NFS3ERR_ISDIR	21	Is a directory. The caller specified a directory in a non-directory operation.
NFS3ERR_INVAL	22	An argument was passed to the z/OS NFS server was not valid.
NFS3ERR_FBIG	27	File too large. The operation caused a file to grow beyond the server's limit.
NFS3ERR_NOSPC	28	No space left on device. The operation caused the server's file system to reach its limit.
NFS3ERR_ROFS	30	Read-only file system. Write tried on a read-only file system.
NFS3ERR_MLINK	31	Too many links.
NFS3ERR_NAMETOOLONG	63	File name too long. The file name in an operation was too long.
NFS3ERR_NOTEMPTY	66	Directory not empty. Tried to remove a directory that was not empty.
NFS3ERR_DQUOT	69	Disk quota exceeded. The client's disk quota on the server has been exceeded.
NFS3ERR_STALE	70	The file handle given in the arguments was not valid. That is, the file referred to by that file handle no longer exists, or access to it has been revoked.
NFS3ERR_NOT_SYNC	10001	File handle is not valid.
NFS3ERR_BAD_COOKIE	10002	Synchronization mismatch on SETATTR.
NFS3ERR_BAD_COOKIE	10003	REaddir and REaddirplus cookie is stale.
NFS3ERR_NOTSUPP	10004	Operation is not supported.
NFS3ERR_TOOSMALL	10005	Buffer or request is too small.
NFS3ERR_SERVERFAULT	10006	Server abandons the request.
NFS3ERR_BADTYPE	10007	Type of an object is not supported.
NFS3ERR_JUKEBOX	10008	Request was initiated, but not completed.

Table 40 on page 141 lists the externalized return codes that are defined by the NFS protocol that is mapped to the z/OS UNIX return codes.

Table 40. z/OS NFS version 1 protocol externalized return code mapping to z/OS UNIX

NFS Protocol Value	Return Code	z/OS UNIX Value	Return Code
NFSERR_PERM	1	EPERM	139
NFSERR_NOENT	2	ENOENT	129
NFSERR_IO	5	EIO	122
NFSERR_NXIO	6	ENXIO	138
NFSERR_ACCESS	13	EACCES	111
NFSERR_EXIST	17	EEXIST	117
NFSERR_NODEV	19	ENODEV	128
NFSERR_NOTDIR	20	ENOTDIR	135
NFSERR_ISDIR	21	EISDIR	123
NFSERR_EINVAL	22	EINVAL	121
NFSERR_FBIG	27	EFBIG	119
NFSERR_NOSPC	28	ENOSPC	133
NFSERR_ROFS	30	EROFS	141
NFSERR_NAMETOOLONG	63	ENAMETOOLONG	126
NFSERR_NOTEMPTY	66	ENOTEMPTY	136
NFSERR_DQUOT	69	EDQUOT	1133
NFSERR_STALE	70	ESTALE	1134

Table 41 contains the return codes that are defined by the NFS version 3 protocol and z/OS UNIX.

Table 41. z/OS NFS return codes between NFS version 3 protocol and z/OS UNIX

NFS version 3 Protocol value	z/OS UNIX value	hex	Description
NFSS3ERR_PERM	EPERM	008B	Operation not permitted
NFS3ERR_NOENT	ENOENT	0081	No such file/directory
NFS3ERR_IO	EIO	007A	I/O error
NFS3ERR_EXIO	ENXIO	008A	No such devices
NFS3ERR_ACCESS	EACCESS	006F	Permission is denied
NFS3ERR_EXIST	EEXIST	0075	The file exists
NFS3ERR_XDEV	EXDEV	0090	Cross device hard link
NFS3ERR_NODEV	ENODEV	0080	No such device
NFS3ERR_NODIR	ENODIR	0087	Not a directory
NFSS3ERR_ISDIR	EISDIR	007B	The file specified is a directory
NFSS3ERR_INVAL	EINVAL	0079	The parameter is incorrect
NFS3ERR_FBIG	EFBIG	0077	The file is to large
NFS3ERR_NOSPC	ENOSPC	0085	No space on device
NFS3ERR_ROFS	EROFS	008D	Read-only file system
NFS3ERR_MLINK	EMLINK	007D	Too many links occurred
NFS3ERR_NAMETOOLONG	ENAMETOOLONG	007E	Filename is too long
NFS3ERR_NOTEMPTY	ENOTEMPTY	0088	Directory not empty

Table 41. z/OS NFS return codes between NFS version 3 protocol and z/OS UNIX (continued)

NFS version 3 Protocol value	z/OS UNIX value	hex	Description
NFS3ERR_DQUOT	EDQUOT	046D	Disk quota exceeded
NFS3ERR_STALE	ESTALE	046E	Stale file handle
NFS3ERR_REMOTE	EREMOTE	046F	Too many levels of remote in path
NFS3ERR_BADHANDLE	EBADF	0071	Bad file handle/descriptor
NFS3ERR_NOT_SYNC	N/A		See note
NFS3ERR_BAD_COOKIE	N/A		See note
NFS3ERR_NOTSUP	ENOSYS	0456	Protocol not supported
NFS3ERR_TOOSMALL	N/A		See note
NFS3ERR_SERVERFAULT	EIO	007A	I/O error
NFS3ERR_BADTYPE	ENXIO	008A	No such device
NFS3ERR_JUKEBOX	EAGAIN	0070	Resource unavailable

**Note:** If z/OS NFS client provides necessary recovery, then there is no corresponding z/OS UNIX return code.

---

## Appendix C. Handling of the file size value

Many NFS procedures (such as `nfs_lookup` and `nfs_getattr`) in the NFS protocol require the file size to be returned. This appendix explains some performance and accuracy considerations in obtaining the file size value.

The meaning of the file size value that is returned by the NFS and how fast the file size is returned depends on the following conditions:

- Whether you use text or binary processing mode
- The type of MVS data set being accessed
- If the data set is system-managed
- If you use **fastfilesize** processing

---

### Storage of the file size value

How the file size value is stored affects how quickly files are accessed and depends on the type of MVS data set used.

### System-managed PS, VSAM, and PDSE data sets

For system-managed PS, VSAM KSDS, VSAM ESDS, VSAM RRDS, and PDSE data sets, the file size value is stored on DASD and is returned quickly.

Text and binary file size are saved on non-volatile storage (DASD) and maintained by the server for these data set types:

- Physical sequential (including striped)
- VSAM ESDS
- VSAM KSDS
- VSAM RRDS
- PDSE members

These data sets must be SMS managed. When the NFS accesses a data set for the first time, it performs a read-for-size to get the text or binary file size and stores this value on DASD. Subsequent file size requests from clients do not cause the server to read for size, thus improving performance. However, when the data set is modified outside the server by a non-NFS application (for example, by the TSO editor), the stored file size could be incorrect. When the data set is accessed again by the server, read-for size must be done to determine the correct file size.

### Migrated system-managed data sets

DFSMS/MVS allows data set attribute accessibility for SMS managed data sets, without having to recall the data set if the data set is migrated under DFSMS/MVS V1R3. Supported data set types are SMS managed PS, VSAM ESDS, VSAM KSDS, VSAM RRDS, PDS, and PDSE. Migrated PDS/PDSE members are not supported.

The z/OS NFS server is able to obtain the attributes of a supported SMS managed migrated data set without recalling the data set. Attributes such as the record format and file size are saved to DASD. Subsequent file size requests do not cause a recall of the supported SMS managed migrated data set, thus improving performance. However, when the data set is modified outside the server by a non-Network File System application (for example, by the TSO editor) before it was migrated, the stored file size could be incorrect. When the data set is accessed again by the server, a recall must be done to determine the correct file size.

## Non-system-managed, PDS, and direct access data sets

The file size value for non-system-managed data sets, PDS members, and direct access (DA) data sets is cached in virtual storage until timeout but not written to DASD. Therefore, for these types of MVS data sets, the file size value is regenerated after the file is closed or after the server is restarted.

---

### How the file size value is generated

When a file is first accessed (for example with **ls -l** or **dir**), usually the entire file is read to determine its size, except for **recfm(f)** or **recfm(fbs)** where the binary size can be computed without reading the file. If the file is a system-managed PS, VSAM, or PDSE member, both binary and text file sizes are stored on DASD, so that subsequent file size requests do not require the file to be read.

Binary file size can be quickly generated by using **recfm(f)** or **recfm(fbs)** to specify a fixed-length record format for the MVS data set. With this format type, the server pads the last logical record with binary zeros in binary mode processing, because MVS always expects complete logical records. If the application tolerates these zeros, using **recfm(f)** or **recfm(fbs)** allows the binary size to be computed quickly because the number of bytes can be computed from the number of blocks, which is stored by MVS.

If you need the exact file size and are using binary mode processing, map it to a variable-format, sequential data set on DASD so that the NFS does not need to pad a partially filled last MVS logical record to a record boundary.

For reading small files or the beginning of files, the read-for-size might not add any processing time. As the file is being read for size, the beginning of the file is stored in the buffers set aside by the **maxrdfsleft** site attribute, until the buffers are full. When the application reads the beginning of the file, this read is fast because it reads directly from the buffer.

MVS stores the number of blocks (rather than the number of bytes) in an MVS file. For most files, therefore, without reading the entire file, the NFS can only give an estimate of the number of bytes in the file, not the exact number of bytes in the file. Even when the server could get the exact byte count without reading the file, the file size could change depending on the file's processing attributes.

For example, selecting **text** mode processing introduces line terminators such as **'lf'**, **'crlf'**, or **'\n'** into the file, thus changing the perceived size of the file. As another example, suppose you select **text** mode processing with blank stripping enabled on a fixed-length record format file. That causes the server to remove trailing blanks from each record, again changing the perceived size of the file. In these examples, when you first request a file, the server must read the entire file to determine its exact size in bytes.

### Using fastfilesize to avoid read-for-xize

If you can use an approximate file size for a PDS, PDSE, DA, or non-system-managed data set, you can specify the **fastfilesize** attribute to improve performance. With this attribute, the server estimates the size without opening and reading the entire file.

**PDS members** - For PDS and PDSE members, the **fastfilesize** attribute gets the file size from SPF statistics if they exist; otherwise, a zero file size is returned.



**DA data sets** - For DA data sets, an approximate file size is calculated based on the device characteristics, the number of disk tracks in use, and the block size of the data set.

**VSAM** - For non-system-managed VSAM data sets, the estimated size using **fastfilesize** is zero. Therefore, **cat** or **vi** won't show any data.

The **fastfilesize** attribute speeds up data set access by calculating approximate file sizes during data set access. Use this only when you are browsing through files (using the **ls** UNIX command for example) because some commands (such as **cp** or **copy**) might not work correctly if **fastfilesize** is set. When reading or modifying a data set, the **nofastfilesize** attribute should be used to ensure accurate results.

## Nofastfilesize

When you use the default, **nofastfilesize** attribute, the NFS reads the entire file or member to get the file size. It stores the file size value in cache until timeout. If the server's default has been changed to **fastfilesize**, you can still use the **nofastfilesize** attribute to override it. For example:

```
$ ls -l "filename,nofastfilesize"
```

Using this attribute might cause a delay when first accessing very large data sets.

**Note:** When directly mounting on a fully qualified data set name and **nofastfilesize** is specified, the server must return the mount size as part of getting the attributes for the mount. This can slow down the completion of the mount command.



---

## Appendix D. Handling of the time stamps

UNIX file attributes define the following time stamps:

***atime*** - The last time the file was accessed (read)

***mtime*** - The last time the file was modified (write)

***ctime*** - The last time the file status was changed (chmod)

The NFS handles time stamps differently for these types of data sets:

- System-managed PS data sets and system-managed VSAM data sets
- Direct Access data sets and non-system-managed PS data sets
- Non-system-managed VSAM data sets
- PDS and PDSE members

---

### Time stamps for system-managed VSAM and PS data sets

For system-managed PS data sets and system-managed VSAM data sets, *atime* and *mtime* are fully maintained, and the *ctime* is set to the *mtime*.

---

### Time stamps for non-system-managed PS and DA data sets

For non-system-managed physical sequential (PS) and direct access (DA) data sets, consider the following conditions:

- How time stamps are stored.
- The requirements of your workstation programs.
- The type of multiple virtual system (MVS) data set used to store the file.

### Storing time stamps

For non-system-managed physical sequential (PS) and direct access (DA) data sets, the Network File System temporarily stores the time stamps in virtual storage, but not on direct access storage device (DASD). These cached attributes are purged when the file times out and closes or when the server is restarted. When the file is accessed again, the time stamps are re-generated.

### Client program requirements

Some workstation-based utilities (such as **make**) rely on date and time stamps to determine whether to recompile. For example, **make** checks the update time of the object file with the source file and recompiles if the source has been updated. Before storing these types of files using the MVS server, examine them before moving them to ensure that these attributes are unimportant. In an environment which relies on such utilities, use system-managed PS data sets.

### Generating time stamps

This is how the NFS generates *atime* and *mtime* for non-system-managed PS and DA data sets from the MVS dates:

```
atime
= mtime = reference_date + time_increment
ctime = creation_date + time_increment
```

*time\_increment* is either the server local time or 23:59 hours. If *reference\_date* or *creation\_date* is equal to the server local date, the server local time is added. Otherwise, a fixed value of 23 hours and 59 minutes is added.

If *reference\_date* = 0 (that is, the file has not yet been referenced), *atime* and *mtime* are set equal to *ctime*.

---

## Time stamps for non-system-managed VSAM data sets

The time stamps for these types of data sets are set to the current time.

---

## Time stamps for PDSs and PDSEs

An MVS PDS data set can act as a UNIX directory. Members of the PDS are files within the UNIX directory. When the directory is accessed by the client, the UNIX times are expected for each file.

Ordinarily, MVS does not maintain time stamps for members of a PDS. The UNIX time stamps here are generated from the MVS creation and reference dates of the PDS data set containing the members. This is how the time stamps for PDS members are generated:

```
atime
= mtime = reference_date + time_increment
ctime = creation_date + time_increment
```

ISPF is an MVS product that does maintain some additional statistics for each member. They include the creation date and the last modification date and time.

If the ISPF time stamps are present for a PDS member, this is how the server generates the time stamps and initializes the UNIX times:

```
atime = mtime = modification_date + modification_time
ctime = ISPF_creation_date + time_increment
```

***time\_increment*** is either server local time or 23:59 hours as described for non-VSAM data sets.

The server also creates new ISPF statistics for PDS members created by the clients. The ISPF statistics are created even if existing members do not have statistics.

The time stamp information is saved in the PDS directory according to ISPF conventions. If STATS=ON was specified when the member was created, the server uses them to get more accurate attributes. Even if STATS=ON was not specified originally, the server writes back new time stamp information if the member is modified from the workstation.

Time stamp generation for a PDSE member is identical to that of a PDS with one exception. Accurate ***mtime*** of a PDSE member is returned to a client as the result of a file attribute request for that PDSE member.

---

## Setting time stamps

NFS clients can issue SETATTR requests to set the *atime* and *mtime* for a system-managed PS or VSAM data set. For PDSE members, setting *mtime* is allowed, but setting *atime* is not supported. PDSE member *mtime* is also maintained by PDSE access methods, so it is modified when a TSO user modifies the PDSE member.



## Appendix E. Sample program for lockd and statd

Figure 22 shows a sample program for record locking and file locking.

```

/*****
/* The System V lockf function has the following values:
/*
/*      #include
/*      int lockf(int fd, int function, long size);
/*
/* where 'function' has one of the following values:
/*      F_ULOCK Unlock a previously lockd region
/*      F_LOCK  Lock a region (blocking)
/*      F_TLOCK Test and lock a region (nonblockig)
/*      F_TEST  Test a region to see if it is lockd
/*
/* The lockf function uses the current file offset ( which the
/* process can set using the lseek system call) and the size argument
/* to define the "record." The record starts at the current offset
/* and extends forward for a positive size, or extends backwards for
/* a negative size. if the size is zero, the record affected extends
/* from the current offset through the largest file offset (the end
/* of file). Doing an lseek to the beginning of the file followed by
/* a lockf with a size of zero locks the entire file.
/*
/* The lockf function provides both the ability to set a lock and to
/* test if a lock is set. When the function is F_LOCK and the region
/* is already locked by another process, the calling process is put
/* to sleep until the region is avaiable. This is termed blocking.
/* The F_TLOCK operation, however, is termed a nonblocking call - if
/* the region is not available, lockf returns immediately with a
/* value of -1 and errno set to either EAGAIN or EACCESS. Also, the
/* F_TEST operation allows a process to test if a lock is set,
/* without setting a lock.
/*
/* NOTE: To do a nonblocking lock, we must use the F_TLOCK
/* operation. If we write
/*
/*      ....
/*      if (lockf(fd, F_TEST, size) == 0) {
/*          rc = lockf(fd, F_LOCK, size);
/*          ...
/*      }
/*      There is a chance the some other process can issue lockf
/*      call between the F_TEST and the F_LOCK, which cause the
/*      F_LOCK to block. The F_TLOCK provides the ability to do
/*      the test and set a single operation, which is needed.
*****/
/* 4.3BSD Advisory Locking
/* The flock system call is provided to lock and unlock a "file"
/*
/*      #include
/*      int flock(int fd, int operation);

```

Figure 22. Sample program for record locking and file locking

```

/*
/* fd is a file descriptor of an open file, and operation is built
/* from the following constants:
/*
/*
/* LOCK_SH Shared lock
/* LOCK_EX Exclusive lock
/* LOCK_UN Unlock
/* LOCK_NB Don't block when locking
/*
/*
/* More than one shared lock can be applied to a file at any time,
/* but a file cannot have both a single lock and an exclusive lock,
/* or multiple exclusive locks at any time. The permissible locking
/* operations are
/*
/* LOCK_SH Shared lock(blocking)
/* LOCK_EX Exclusive lock(blocking)
/* LOCK_SH | LOCK_NB Shared lock(nonblocking)
/* LOCK_EX | LOCK_NB Exclusive lock(nonblocking)
/*
/*
/* If the lock was successful, zero is returned by flock, otherwise
/* -1 is returned. If the lock fails because the file is already
/* locked and a nonblocking lock was requested (LOCK_NB), the global
/* errno is set to EWOULDBLOCK.
/*
/*
/* NOTE: must use "-lbsd" argument with the "cc" command in order
/* to obtain the flock function for AIX
/*
/*****
#define F_ULOCK 0 /* Unlock locked sections */
#define F_LOCK 1 /* Lock a section for exclusive use */
#define F_TLOCK 2 /* Test and lock a section for
/* exclusive use */
#define F_TEST 3 /* Test section for locks by other
/* processes */
main(int argc, char **argv)
{int fd, j;
long tloc;
unsigned long seconds = 10;
pid_t pid;
int status;
if (argc < 2) {
fprintf(stderr, "arg1 = file to lock.\narg2 = seconds to hold lock "
"(default = 10 seconds).\n");
return 101;}
if (argc > 2) { /* Parse seconds value. */
if (1 != sscanf(argv[2], "%ld", &seconds)) {
fprintf(stderr, "%s: cannot understand arg2 value (%s).\n",
argv[0], argv[2]);
return 109;} }

```



```

time(&tloc);          /* Get current time. */
printf("Starting %s on file \"%s\" at %s",
argv[0], argv[1], asctime(localtime(&tloc)));
fd = open(argv[1], O_RDWR);
if (fd < 0) {
    perror("open failed");
    return 102;}
printf("%s: calling flock [LOCK_SH].\n", argv[0]);
j = flock(fd, LOCK_SH);
if (j) {
    perror("flock failed");
    return 103;}
printf("%s: flock successful; sleeping for %d seconds...\n", \
    argv[0], seconds);
j = sleep(seconds);
if (!j) fprintf(stderr, "%s: exit code %d from call to sleep for %d "
    "seconds.\n", argv[0], j, seconds);

printf("%s: calling flock [LOCK_UN].\n", argv[0]);
j = flock(fd, LOCK_UN);
if (j) {
    perror("flock failed");
    return 103;}
else printf("%s: flock successful.\n", argv[0]);

printf("%s: calling flock [LOCK_EX].\n", argv[0] );
j = flock(fd, LOCK_EX);
if (j) {
    perror("flock failed");
    return 103;}

printf("%s: flock successful; sleeping for %d seconds...\n", \
    argv[0], seconds);

j = sleep(seconds);
if (!j) fprintf(stderr, "%s: exit code %d from call to sleep for %d "
    "seconds.\n", argv[0] , j, seconds);

printf("%s: calling flock [LOCK_UN] .\n", argv[0] );
j = flock(fd, LOCK_UN);
if (j) {
    perror("flock failed");
    return 103;}
else printf("%s: flock successful.\n", argv[0]);

if (-1 == lseek(fd, 0, SEEK_SET)) {
    perror("lseek to 0 error");
    goto end;}
printf("%s: calling lockf [lock at 0 for 3].\n", argv[0]);

```

```

if (lockf(fd, F_LOCK, 3)) {
    perror("lockf error");
    goto end;}
if (-1 == lseek(fd, 10, SEEK_SET)) {
    perror("lseek to 10 error");
    goto end;}
printf("%s: calling lockf [flock at 10 for 3] .\n", argv[0]);
if (lockf(fd, F_LOCK, 3)) {
    perror("lockf error");
    goto end;}
pid = fork();
if (pid == 0) { /* New process; test locks. */
    if (lockf(fd, F_TEST, 3) == 0) {
        printf("%s: lock test from new process succeeded when it ought "
            "to have failed.\n", argv[0]);
        return 1;}
    else {
        perror("New process correctly finds lock not available");
        return 0;} } /* End of child process. */
else { /* Parent process. */
    if (wait(&status) != pid) {
        perror("Wait failed");
        goto end;} }

    if (lockf(fd, F_ULOCK, 3)) fprintf(stderr, "%s: unlock at 10 for 3 "
        "failed (errno = %d).\n", argv[0] , errno);
    if (-1 == lseek(fd, 0, SEEK_SET)) {
        perror("lseek to 0 error");
        goto end;}
    if (lockf(fd, F_ULOCK, 3)) fprintf(stderr, "%s: unlock at 0 for 3 "
        "failed (errno = %d).\n", argv[0], errno);
    end:  time(&tloc); /* Get current
time. */ printf("Ending %s at %s", argv[0] ,
asctime(localtime(&tloc))); close(fd);

```

---

## Notices

This information was developed for products and services offered in the U.S.A. IBM may not offer the products, services, or features discussed in this document in other countries. Consult your local IBM representative for information on the products and services currently available in your area. Any reference to an IBM product, program, or service is not intended to state or imply that only that IBM product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any IBM intellectual property right may be used instead. However, it is the user's responsibility to evaluate and verify the operation of any non-IBM product, program, or service.

IBM may have patents or pending patent applications covering subject matter described in this document. The furnishing of this document does not give you any license to these patents. You can send license inquiries, in writing, to:

IBM Director of Licensing  
IBM Corporation  
North Castle Drive  
Armonk, NY 10504-1785  
U.S.A.

For license inquiries regarding double-byte (DBCS) information, contact the IBM Intellectual Property Department in your country or send inquiries, in writing, to:

IBM World Trade Asia Corporation Licensing  
2-31 Roppongi 3-chome, Minato-ku  
Tokyo 106, Japan

**The following paragraph does not apply to the United Kingdom or any other country where such provisions are inconsistent with local law:**

INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Some states do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement may not apply to you.

This information could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the publication. IBM may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time without notice.

IBM may use or distribute any of the information you supply in any way it believes appropriate without incurring any obligation to you.

Licensees of this program who wish to have information about it for the purpose of enabling: (i) the exchange of information between independently created programs and other programs (including this one) and (ii) the mutual use of the information which has been exchanged, should contact:

IBM Corporation  
Information Enabling Requests  
Dept. DZWA  
5600 Cottle Road  
San Jose, CA 95193 U.S.A.

Such information may be available, subject to appropriate terms and conditions, including in some cases, payment of a fee.

The licensed program described in this information and all licensed material available for it are provided by IBM under terms of the IBM Customer Agreement, IBM International Program License Agreement, or any equivalent agreement between us.

---

## Programming interface information

This publication documents intended Programming Interfaces that allow the customer to write programs to obtain the services of z/OS DFSMS.

---

## Trademarks

The following terms are trademarks of the IBM Corporation in the United States, or other countries, or both:

AIX	MVS/ESA
DFSMS/MVS	OpenEdition
DFSMSHsm	OS/2
IBM	OS/390
IBMLink	RACF
Language Environment	Resource Link
MVS	RS/6000
MVS/DFP	z/OS

Java and all Java-based trademarks are trademarks of Sun Microsystems, Inc. in the United States, other countries, or both.

Microsoft, Windows, Windows NT, and the Windows logo are trademarks of Microsoft Corporation in the United States, other countries, or both.

UNIX is a registered trademark of The Open Group in the United States and other countries

Other company, product, and service names, which may be trademarks or service marks of others.

---

# Glossary

This glossary defines technical terms and abbreviations used in DFSMS documentation. If you do not find the term you are looking for, refer to the index of the appropriate DFSMS manual or view the *IBM Dictionary of Computing Terms* located at:

<http://www.ibm.com/networking/nsg/nsgmain.htm>

This glossary includes terms and definitions from:

- The *American National Standard Dictionary for Information Systems*, ANSI X3.172-1990, copyright 1990 by the American National Standards Institute (ANSI). Copies may be purchased from the American National Standards Institute, 11 West 42nd Street, New York, New York 10036. Definitions are identified by the symbol (A) after the definition.
- The *Information Technology Vocabulary* developed by Subcommittee 1, Joint Technical Committee 1, of the International Organization for Standardization and the International Electrotechnical Commission (ISO/IEC JTC1/SC1). Definitions of published part of this vocabulary are identified by the symbol (I) after the definition; definitions taken from draft international standards, committee drafts, and working papers being developed by ISO/IEC JTC1/SC1 are identified by the symbol (T) after the definition, indicating that final agreement has not yet been reached among the participating National Bodies of SC1.
- The *IBM Dictionary of Computing*, New York: McGraw-Hill, 1994.

The following cross-reference is used in this glossary:

**See:** This refers the reader to (a) a related term, (b) a term that is the expanded form of an abbreviation or acronym, or (c) a synonym or more preferred term.

## A

**access.** To obtain computing services.

**access method.** (1) A mainframe data management routine that moves data between storage and an I/O device in response to requests made by a program. (2) The part of the distributed data management architecture which accepts commands to access and process the records of a file.

**access permission.** A group of designations that determine who can access a particular AIX or UNIX file and how the user can access the file.

**ACS.** See *automatic class selection*.

**ACS routine.** A procedural set of ACS language statements. Based on a set of input variables, the ACS language statements generate the name of a predefined SMS class, or a list of names of predefined storage groups, for an MVS file.

**address.** The unique identifier assigned to each device or workstation connected to a network.

**address space.** The complete range of addresses in memory available to a computer program.

**Advanced Interactive Executive (AIX).** IBM's licensed version of the UNIX operating system.

**AIX.** See *Advanced Interactive Executive*.

**alias.** An alternative name for an ICF user catalog, a non-VSAM file, or a member of a partitioned data set (PDS) or PDSE.

**alias entry.** An entry that relates an alias to the real entry name of a user catalog or nonVSAM data set.

**allocation.** (1) Generically, the entire process of obtaining a volume and unit of external storage, and setting aside space on that storage for a data set. (2) The process of connecting a program to a data set or devices.

**American Standard Code for Information Interchange (ASCII).** The standard code used for information exchange among data processing systems, data communication systems, and associated equipment. It uses a coded character set consisting of 7-bit coded characters.

**APAR.** See *authorized program analysis report*.

**APF.** Authorized program facility.

**API.** See *application programming interface*.

**application programming interface (API).** A formally defined programming language interface between an IBM system control program or a licensed program and the user of a program.

**ASCII.** See *American National Standard Code for Information Interchange*.

**atime.** The time when the file was last accessed.

**automatic class selection (ACS).** A mechanism for assigning Storage Management Subsystem classes and storage groups to data sets.

**automatic class selection (ACS) routine.** A procedural set of ACS language statements. Based on a set of input variables, the ACS language statements generate the name of a predefined SMS class, or a list of names of predefined storage groups, for a data set.

**authorized program analysis report (APAR).** A request for correction of a problem caused by a suspected defect in a current unaltered release of a program.

## B

**basic sequential access method (BSAM).** An access method for storing or retrieving data blocks in a continuous sequence, using either a sequential access or a direct access device.

**BIOD.** The caching daemon that caches directory lookups and file data when remote files are accessed from the host.

**block.** A string of data elements recorded, processed, or transmitted as a unit. The elements can be characters, words, or physical records.

**BSAM.** See *Basic sequential access method*.

## C

**CCSID.** See *Coded Character Set Identifier*.

**CDRA.** Character data representation architecture.

### Character Data Representation Architecture (CDRA)

**API.** A set of identifiers, services, supporting resources, and conventions for consistent representation, processing, and interchange of character data.

**client.** (1) A user. (2) A consumer of resources or services. (3) A functional unit that receives shared services from a server. (4) A system that is dependent on a server to provide it with programs or access to programs. (5) On a network, the computer requesting services or data from another computer.

**client-server relationship.** Any process that provides resources to other processes on a network is a *server*. Any process that employs these resources is a *client*. A machine can run client and server processes at the same time.

**Coded Character Set Identifier (CCSID).** A 16-bit number that identifies a specific encoding scheme identifier, character set identifiers, code page identifiers,

and additional coding required information. The CCSID uniquely identifies the coded graphic character representation used.

**connection.** An association established between functional units for conveying information.

**current directory.** The currently active directory; the directory that is searched when you enter a file name without indicating the directory that contains the file name. When you specify a file name without specifying a directory, the system assumes that the file is in the current directory.

## D

**daemon.** A background process that is initiated at system start that continuously performs a function required by another process.

**DA.** Direct access.

**DAE.** Dump Analysis and Elimination.

**DAM.** Direct access method.

**DASD.** Direct access storage device.

**DASD volume.** A Direct Access Storage Device space identified by a common label and accessed by a set of related addresses. See also *volume*, *primary storage*, *migration level 1*, *migration level 2*.

**data control block (DCB).** A control block used by access method routines in storing retrieving data.

**data set.** In DFSMS, the major unit of data storage and retrieval, consisting of a collection of data in one of several prescribed arrangements and described by control information to which the system has access. In z/OS non-UNIX environments, the terms *data set* and *file* are generally equivalent and sometimes are used interchangeably. See also *file*. In z/OS UNIX environments, the terms *data set* and *file* have quite distinct meanings.

**data set organization.** The type of arrangement of data in a data set. Examples are sequential organization or partitioned organization.

**DBCS.** Double byte character set.

**DCB.** See *data control block*.

**DES authentication.** Requires a client to send credentials (its name, conversation key, window key, and a time stamp) to the server. The server then returns a verifier to the client. Data Encryption Standard (DES) credentials are sometimes called "secure" credentials because they are based on a sender's ability to encrypt data using a common time reference. DES-style credentials require a randomly generated key used to encrypt a common reference time which is then used in

turn to create a *conversation key*. DES-style credentials are less vulnerable to penetration than UNIX-style credentials because of the generated conversation key and the data that is encrypted with the conversation key. Other forms of attack (for example, replaying message traffic) are resisted because of the timestamp in the transmission. The Network File System does not support DES-style credentials.

**DFSMSdfp.** A DFSMS functional component or base element of z/OS, that provides functions for storage management, data management, program management, device management, and distributed data access.

**direct access file.** A type of MVS file for storing data on a random access device that is accessed using a record address.

**directory.** A file that maps the names of other directories and files to their locations.

**DOS.** Disk operating system.

**DSCB .** Data set control block.

## E

**EBCDIC.** See *extended binary-coded decimal interchange code*.

**end user.** A person in a data processing installation who requires the services provided by the computer system.

**EOR.** End of record.

**entry-sequenced data set (ESDS).** In VSAM, a data set whose records are loaded without respect to their contents and whose RBAs cannot change. Records are retrieved and stored by addressed access, and new records are added at the end of the data set.

**ESDS.** See *entry-sequenced data set (ESDS)*.

**ESS.** Enterprise Storage Server.

**ETR.** Electronic technical response.

**exports data set.** An MVS file on the server containing entries for directories that can be exported to Network File System clients. It is used by the server to determine which MVS files and prefixes can be mounted by a client, and to write-protect MVS files on the server.

**extended binary-coded decimal interchange code (EBCDIC).** A coded character set consisting of 8-bit coded characters.

**External Data Representation (XDR).** A standard developed by Sun Microsystems, Incorporated for representing data in machine-independent format. XDR

is a vendor independent way of representing the data. By using the XDR standard data representation convention, systems do not have to understand and translate every data format that exists on the network; there is only the one convention. Data is translated into XDR format before it is sent over the network and, at the reception point, is translated into the data convention used there. This means that new computer architectures can be integrated into the network without requiring the updating of translation routines. The new architecture simply includes a routine that translates its data format into XDR format and the new member of the network is ready to go.

Using XDR, data can be accessed or exchanged among machines of various hardware and software architectures without any translation or interpretation problems. Word lengths, byte ordering, and floating point representations appear to be the same to all nodes in the network.

## F

**FDDI.** Fiber Distributed Data Interface (FDDI).

**file.** A collection of information treated as a unit. In non-UNIX System Services MVS environments, the terms *data set* and *file* are generally equivalent and are sometimes used interchangeably. See also *data set*.

**file handle.** A file handle is used by the client and server sides of the Network File System to specify a particular file or prefix.

**file system.** In the z/OS UNIX HFS environment, the collection of files and file management structures on a physical or logical mass storage device, such as a diskette or minidisk. See also *HFS data set*.

**File Transfer Protocol (FTP).** A TCP/IP protocol used for transferring files to and from foreign hosts. FTP also provides the capability to access directories.

**FMID.** Function modification identifier.

**FUB.** File usage block.

**foreign host.** Any host on the network other than the local host.

**free space.** Space reserved within the control intervals of a key-sequenced data set for inserting new records into the data set in key sequence or for lengthening records already there; also, whole control intervals reserved in a control area for the same purpose.

**FTP.** File Transfer Protocol.

## G

**gateway.** A functional unit that interconnects two computer networks with the different network

architectures. A gateway connects networks or systems of different architectures. A bridge interconnects networks or systems with the same or similar architectures.

**GID.** See *group number (GID)*.

**group.** (1) With respect to partitioned data sets, a member and the member's aliases that exist in a PDS or PDSE, or in an unloaded PDSE. (2) A collection of users who can share access authorities for protected resources.

**group number (GID).** A unique number assigned to a group of related users. The group number can often be substituted in commands that take a group name as an argument.

**GXB .** Global exit block.

## H

**handle.** (1) In the Advanced DOS and OS/2 operating systems, a binary value created by the system that identifies a drive, directory, and file so that the file can be found and opened. (2) In the AIX operating system, the data structure that is a temporary local identifier on an object. Allocating a handle creates it. Binding a handle makes it identify an object at a specific location. (3) In the OS/400 application programming interfaces, a variable that represents an object.

**HFS.** See *hierarchical file system (HFS)*.

**hierarchical file system (HFS) data set.** A data set that contains a POSIX-compliant hierarchical file system, which is a collection of files and directories organized in a hierarchical structure, that can be accessed using the UNIX System Services MVS facilities.

**host.** A computer connected to a network that provides an access method to that network. A host provides end-user services.

## I

**ICF.** See *integrated catalog facility (ICF)*.

**ICP.** Interconnect Control Program (ICP).

**IDCAMS.** Integrated catalog access methods services.

**integrated catalog facility (ICF).** In the Data Facility Product (DFP), a facility that provides for integrated catalog facility catalogs.

**Internet.** A specific internetwork that includes ARPANET, MILNET, and NSFnet. These networks use the TCP/IP protocol suite. See also *internetwork*.

**Internet Protocol (IP).** The TCP/IP layer between the higher-level host-to-host protocol and the local network protocols. IP uses local area network protocols to carry packets in the form of diagrams to the next gateway or destination host.

**internetwork.** A collection of packet-switched networks that are connected by gateways. They work as a single network.

**interoperability.** The ability of hardware and software from different vendors to communicate on a network.

**inter-process communication (IPC).** Ways for programs to communicate data to each other and to synchronize their activities. Semaphores, signals, and internal message queues are common methods of inter-process communication.

**IP.** Internet Protocol.

**IPC.** See *inter-process communication (IPC)*.

## J

**JCL.** See *job control language*.

**Job control language (JCL).** A problem-oriented language used to identify the job or describe its requirements to an operating system.

## K

**key-sequenced data set (KSDS).** A VSAM data set whose records are loaded in key sequence and controlled by an index. Records are retrieved and stored by keyed access or by addressed access, and new records are inserted in the data set in key sequence because of free space allocated in the data set. Relative byte addresses of records can change because of control interval or control area splits.

**KSDS.** See *key-sequenced data set (KSDS)*.

## L

**LFS.** Logical file system.

**local host.** The computer to which a user's terminal is directly connected.

**local network.** That portion of a network physically connected to the host without intermediate gateways.

## M

**management class.** A collection of management attributes, defined by the storage administrator, used to control the release of allocated but unused space; to control the retention, migration, and backup of data



sets; to control the retention and backup of aggregate groups, and to control the retention, backup, and class transition of objects.

**master catalog.** A key-sequenced data set or file with an index containing extensive data set and volume information that VSAM requires to locate data sets or files, to allocate and deallocate storage space, to verify the authorization of a program or operator to gain access to a data set or file, and to accumulate usage statistics for data sets or files.

**MBCS.** Multiple byte character set.

**migration level 1.** DFSMSHsm-owned DASD volumes that contain data sets migrated from primary storage volumes. The data can be compressed. See also *storage hierarchy*. Contrast with *primary storage*, *migration level 2*.

**migration level 2.** DFSMSHsm-owned tape or DASD volumes that contain data sets migrated from primary storage volumes or from migration level 1 volumes. The data can be compressed. See also *storage hierarchy*. Contrast with *primary storage*, *migration level 1*.

**mkdir.** An AIX, UNIX, OS/2 and DOS command used to make a new directory.

**mount.** A host-linked operation which results in a tape cartridge being physically inserted into a tape drive.

**mount handle data set.** A data set used to store the file handles of mount points.

**mount point.** A directory established in a workstation or a server local directory that is used during the transparent accessing of a remote file.

**mtime.** The time of last data modification.

**MTU.** Maximum transmission unit (MTU).

**MVS.** Multiple virtual system.

**MVS/ESA.** Multiple Virtual Storage/Enterprise Systems Architecture. A z/OS operating system environment that supports ESA/390.

**MVS/ESA SP.** An IBM licensed program used to control the z/OS operating system. MVS/ESA SP together with DFSMS compose the base MVS/ESA operating environment. See also *z/OS*.

**mvslogin.** A client command to connect your workstation to MVS.

**mvslogout (or mvslogut).** A client command to break the connection between your workstation and MVS.

## N

**network.** (1) An arrangement of nodes and connection branches. (2) A configuration of data processing devices and software connected for information interchange.

**NFS.** See *z/OS Network File System*.

**NLM.** Network Lock Manager.

**NSM.** Network Status Manager.

**null credentials.** Null credentials are usually associated with diskless workstations where it is impossible to obtain identifying information because there is no local (that is, local to the workstation) repository of information.

## O

**object storage hierarchy.** A hierarchy consisting of objects stored in DB2 table spaces on DASD, on optical or tape volumes that reside in a library, and an optical or tape volumes that reside on a shelf. See also *storage hierarchy*.

**optical volume.** Storage space on an optical disk, identified by a volume label. See also *volume*.

**OpenEdition MVS.** See *z/OS UNIX System Services*.

**OpenEdition MVS file system.** See *z/OS UNIX file system*.

**optical volume.** Storage space on an optical disk, identified by a volume label. See also *volume*.

**Operating System/2 (OS/2).** An operating system used in IBM PC AT, PS/2, and compatible computers.

**OSA2.** Open System Adapter 2 (OSA2).

## P

**partitioned data set (PDS).** A data set on direct access storage that is divided into partitions, called members, each of which can contain a program, part of a program, or data.

**partitioned data set extended (PDSE).** A system-managed data set that contains an indexed directory and members that are similar to the directory and members of partitioned data sets. A PDSE can be used instead of a partitioned data set.

**PDS.** See *partitioned data set (PDS)*.

**PDS directory.** A set of records in a partitioned data set (PDS) used to relate member names to their locations on a DASD volume.

**PDSE.** See *partitioned data set extended (PDSE)*.

**permission code.** A three-digit octal code, or a nine-letter alphabetic code, indicating the access permissions. The access permissions are read, write, and execute.

**permission field.** One of the three-character fields within the permissions column of a directory listing indicating the read, write, and run permissions for the file or directory owner, group, and all others.

**PFS.** Physical file system.

**portable operating system interface (POSIX).** Portable operating system interface for computer environments. An IEEE operating system standard, closely related to the UNIX system (software writing).

**port.** (1) An access point for data entry or exit. (2) A receptacle on a device to which a cable for another device is attached.

**Portmapper.** A server that converts RPC program numbers into port numbers acceptable to the protocol. This server must be running to make RPC calls.

**POSIX .** See *portable operating system interface (POSIX)*.

**primary storage.** A DASD volume available to users for data allocation. The volumes in primary storage are called primary volumes. See also *storage hierarchy*. Contrast with *migration level 1*, *migration level 2*.

**protocol.** (1) A set of semantic and syntactic rules that determines the behavior of functional units in achieving communication. (2) A specification for the format and relative timing of information exchanged between communicating parties.

**PS.** Physical sequential.

**PTF.** Program temporary fix.

## Q

**QSAM.** See *queued sequential access method (QSAM)*.

**queued sequential access method (QSAM).** An extended version of the basic sequential access method (BSAM). Input data blocks awaiting processing or output data blocks awaiting transfer to auxiliary storage are queued on the system to minimize delays in I/O operations.

## R

**RACF.** See *Resource Access Control Facility (RACF)*.

**RBA.** Relative byte address.

**relative byte address (RBA).** The displacement of a data record or a control interval from the beginning of the data set to which it belongs; independent of the manner in which the data set is stored.

**relative record data set (RRDS).** A type of VSAM data set whose records have fixed or variable lengths, and are accessed by relative record number.

**remote procedure call (RPC).** RPC is an independent set of functions used for accessing remote nodes on a network. Using RPC network services, applications can be created in much the same way a programmer writes software for a single computer using local procedure calls. The RPC protocols extend the concept of local procedure calls across the network, which means that distributed applications can be developed for transparent execution across a network.

**Resource Access Control Facility (RACF).** An IBM licensed program that is included in z/OS Security Server and is also available as a separate program for the z/OS and VM environments. RACF provides access control by identifying and verifying the users to the system, authorizing access to protected resources, logging detected unauthorized attempts to enter the system, and logging detected accesses to protected resources.

**Resource Measurement Facility (RMF).** An IBM licensed program or optional element of z/OS, that measures selected areas of system activity and presents the data collected in the format of printed reports, system management facilities (SMF) records, or display reports. Use RMF to evaluate system performance and identify reasons for performance problems.

**rmdir.** An AIX, UNIX, OS/2 and DOS command used to remove (delete) a directory.

**RMF.** See *Resource Management Facility (RMF)*.

**root.** The user name for the system user with the most authority.

**RPC.** Remote procedure call.

**RRDS.** Relative record data set.

## S

**SAF.** System Authorization Facility (SAF).

**SDSF.** System Display and Search Facility (SDSF).

**sequential file.** A type of MVS file that has its records stored and retrieved according to their physical order within the file. It must be on a direct access volume.

**server.** (1) A functional unit that provides shared services to workstations over a network; for example, a file server, a print server, a mail server. (2) On a

network, the computer that contains the data or provides the facilities to be accessed by other computers in the network. (3) A program that handles protocol, queuing, routing, and other tasks necessary for data transfer between devices in a computer system.

**sharing.** A term used in a computing environment to refer to utilizing a file on a remote system. It is done by mounting the remote file system, then reading or writing files in that remote system.

**shell prompt.** The character string on an AIX or UNIX command line indicating the system can accept a command (typically the \$ character).

**showattr.** A client command used to display the values of the site, processing, and data set creation attributes.

**SMF.** See *system management facilities (SMF)*.

**SMP/E.** See *System Modification Program/Extended (SMP/E)*.

**SMS.** See *Storage Management Subsystem (SMS)*.

**SPF.** System productivity facility.

**SRM.** System Resources Manager (SRM).

**SSF.** Software support facility.

**stale file handle.** A file handle is stale when the file handle for a file or prefix is no longer valid.

**stateless.** A stateless server can function correctly without maintaining any protocol state information about any of its clients. The NFS protocol is intended to be as stateless as possible. This avoids complex crash recovery; a client just resends requests until a response is received. The stateless protocol is chosen to minimize the probability of data losses due to a server crash.

**storage hierarchy.** An arrangement of storage devices with different speeds and capacities. The levels of the storage hierarchy include main storage (memory, DASD cache), primary storage (DASD containing uncompressed data), migration level 1 (DASD containing data in a space-saving format), and migration level 2 (tape cartridges containing data in a space-saving format). See also *primary storage*, *migration level 1*, *migration level 2*, *object storage hierarchy*.

**Storage Management Subsystem (SMS).** A DFSMS facility used to automate and centralize the management of storage. Using SMS, a storage administrator describes data allocation characteristics, performance and availability goals, backup and retention requirements, and storage requirements to the system through data class, storage class, management class, storage group, and ACS routine definitions.

**system management facilities (SMF).** A component of z/OS that collects input/output (I/O) statistics, provided at the data set and storage class levels, which helps you monitor the performance of the direct access storage subsystem.

**superuser.** The user who can operate without the restrictions designed to prevent data loss or damage to the system (User ID 0).

**System Modification Program/Extended.** Basic tool for installing software changes in programming systems. It controls these changes at the element (module or macro) level, which helps protect system integrity.

## T

**tape volume.** Storage space on a tape, identified by a volume label, which contains data sets or objects and available free space. A tape volume is the recording space on a single tape cartridge or reel. See also *volume*.

**task control block (TCB).** An MVS control block that describes an asynchronous process that is called a task.

**TCB.** See *task control block (TCB)*.

**TCP/IP.** See *Transmission Control Protocol/Internet Protocol (TCP/IP)*.

**TIOT.** Task input/output table.

**transmission control block (TCB).** An internal control block within the TCP/IP address space.

**Transmission Control Protocol (TCP).** A stream communication protocol that includes error recovery and flow control.

**Transmission Control Protocol/Internet Protocol (TCP/IP).** The two fundamental protocols of the Internet protocol suite. The abbreviation TCP/IP is frequently used to refer to this protocol suite. TCP/IP provides for the reliable transfer of data, while IP transmits the data through the network in the form of datagrams. Users can send mail, transfer files across the network, or execute commands on other systems.

**TSO.** Time-sharing option.

**TSO/E.** Time-sharing option extended.

## U

**UCB.** Unit control block.

**UDP.** User datagram protocol.

**UID.** The DFSMSHsm authorized-user identification in 1 to 7 characters. See also *user number (UID)*.

**Unicode Standard.** A universal character encoding standard that supports the interchange, processing, and display of text that is written in any of the languages of the modern world. It can also support many classical and historical texts and is continually being expanded. The Unicode Standard is compatible with ISO/IEC 10646.

**UNIX.** A highly portable operating system originally developed by Bell Laboratories that features multiprogramming in a multi-user environment. UNIX is implemented in the C language. UNIX was originally developed for use on minicomputers but has been adapted on mainframes and microcomputers. It is especially suitable for multiprocessor, graphics, and vector-processing systems.

**UNIX authentication.** Requires a client process to send credentials (the client's machine-name, uid, gid, and group-access-list) to the server.

**user catalog.** An optional catalog used in the same way as the master catalog and pointed to by the master catalog. It lessens the contention for the master catalog and facilitates volume portability.

**User Datagram Protocol (UDP).** A connectionless datagram protocol that requires minimal overhead, but does not guarantee delivery.

**user number (UID).** In the AIX operating system, a number that uniquely identifies a user to the system. It is the internal number associated with a user ID.

## V

**virtual storage access method (VSAM).** An access method for direct or sequential processing of fixed and variable-length records on direct access storage devices. The records in a VSAM file can be organized in logical sequence by a key field (key sequence), in the physical sequence in which they are written on the file (entry sequence), or by relative record number.

**volume.** The storage space on DASD, tape, or optical devices, which is identified by a volume label. See also *DASD volume*, *optical volume*, *tape volume*.

**VRA.** Variable Recording Area (VRA).

**VSAM.** See *virtual storage access method (VSAM)*.

**VSCR.** Virtual storage constraint relief.

## X

**XDR.** See *External Data Representation (XDR)*.

## Z

**z/OS.** z/OS is a network computing-ready, integrated operating system consisting of more than 50 base elements and integrated optional features delivered as a configured, tested system. See also *MVS/ESA SP*.

**z/OS Network File System.** A base element of z/OS, that allows remote access to z/OS host processor data from workstations, personal computers, or any other system on a TCP/IP network that is using client software for the Network File System protocol.

**z/OS UNIX System Services file system.** In the z/OS UNIX environment, the collection of files and file management structures on a physical or logical mass storage device, such as a diskette, minidisk, or a disk drive. See also *hierarchical file system (HFS) data set*.

**z/OS UNIX System Services (z/OS UNIX).** The set of functions provided by the SHELL and UTILITIES, kernel, debugger, file system, C/C++ Run-Time Library, Language Environment, and other elements of the z/OS operating system that allow users to write and run application programs that conform to UNIX standards.

---

# Index

## A

### ABEND

- B37 111
- D37 111
- E37 111

abytes value 18

access method services error message GFSA847I 109

accessing MVS data sets

- command syntax for AIX 39
- command syntax for DFSMS/MVS 53

accessing MVS files

- changing attributes 43
- changing mount point attributes 43
- displaying attributes 45
- getting authorization 44, 61
- line terminators 13
- mount command 43, 57
- mvslogin command 56
- showattr command 44, 67

accessing z/OS UNIX System Services 33

acdirmax attribute 71

acdirmax keyword 128

acdirmin attribute 71

acdirmin keyword 128

acregmax attribute 71

acregmax keyword 128

acregmin attribute 71

acregmin keyword 128

active clients 113

afiles value 18

AIX client, Network File System 39

AIX command syntax 39

AIX commands 40

alias 22, 136

alias name, message GFSA817I 107

American National Standard Code for Information

Interchange (ASCII) 12, 30

ARC0075E DFSMSHsm message 95

ASCII to EBCDIC conversion 79

async attribue 32

async attribute 32

async processing attribute 82

asynchronous daemon task 132

atime 147, 149

AttrCaching attribute 71

attributes

- async 32
- attrtimeout 82, 99
- blankstrip 12
- bufhigh 95, 97, 130
- changing 43
- cln\_ccsid(n) 84
- data set 4
- data set creation 43, 76
- default, displaying 45
- dsntype 100
- dsntype(pds) 24

attributes (*continued*)

- dsorg 98
- dsorg(da) 23
- dsorg(ps) 22
- exports 5
- extlink 32
- fileextmap 15
- for z/OS UNIX System Services 75
- hfs 32
- logicalcache 97
- logout 42
- maplower 9, 12
- maxtimeout 99, 100
- mintimeout 99, 100
- mount point, displaying 45
- nfstasks 101
- noattrtimeout 100
- nofileextmap 15
- nomaplower 9, 12
- nomaxtimeout 100
- noreadtimeout 100
- noretrieve 13
- nowritetimeout 100
- processing 43, 78
- readtimeout 82, 99
- recfm 100
- restimeout 44
- retrieve 13
- saf 41
- safexp 5, 41
- security 5, 41
- sidefile(dsname) 15
- site 86
- srv\_ccsid(n) 84
- sync, async 32
- text 101
- vol 98
- writetimeout 82, 99

attrtimeout attribute 82, 99

attrtimeout processing attribute 78

authentication error 56, 108

restart processing 42

automatic logout, recovering 44, 62

automount facility 57

## B

B37 ABEND 111

basic sequential access method (BSAM) 1, 72

binary files 12

binary processing attribute 79

binary processing mode 12

Biod attribute 71

blanks, handling of 12

blankstrip 136

blankstrip attribute 12

blankstrip processing attribute 79

blks data set attribute 76

- blksize data set attribute 76
- BSAM, QSAM, and VSAM ESDS access to remote files 59
- bufhigh attribute 95, 97, 130
- Bufhigh attribute 71
- bufhigh site attribute 45, 87
- Bufhigh site attribute 68

## C

- C function, fopen 129
- C function, text1 130
- cachewindow site attribute 45, 68, 87
- cachewindow() 11
- case\_insensitive value 18, 36
- case\_preserving value 18, 36
- cat command 10, 12, 24, 25
- cataloged data sets 3, 15
- CCSID 297 58
- CCSID 437 58
- changing
  - attributes 43
  - attributes for mount points 43
  - your MVS password 41, 55
- changing attributes 57
- channel status word, message GFSA876I 111
- Character Data Representation Architecture (CDRA) 58, 131
- checking UNIX permission bits 90
- checklist data set 6
- checklist site attribute 87
- chmod command 10
- chown\_restricted value 18, 36
- client, z/OS NFS introduction 1
- client attributes 71, 73
- client log daemon 131
- client messages 118
- client program requirements 147
- client-server relationship 1
- client transport handle 117
- clients, supported 6
- cln\_ccsid 84, 102, 128
- cln\_ccsid attribute 71
- cln\_ccsid(n) attribute 84
- cln\_ccsid(n) processing attribute 79
- cln\_ccsid parameter 8, 58, 59, 73
- clnt\_call 117
- clnt\_pcreateerror 116
- clnt\_perror message 117
- clntudp\_create 116
- code
  - return code mapping 140
  - return codes 139
- Coded Character Set Identifier (CCSID) 8, 58, 71, 73, 84, 86, 102, 128
- command
  - cat 10, 12, 24, 25
  - chmod 10
  - copy 26
  - cp 21
  - dir 137

- command (*continued*)
  - installing 55
  - ln 34
  - ls 25
  - ls (UNIX) 79, 80, 144
  - mkdir 16, 21, 24, 25
  - modify 43, 44, 101, 102, 104, 105, 112
  - more 31
  - mount 3, 13, 15, 21, 22, 43, 49, 53, 57, 125
    - changing default and mount point attributes 76
    - changing mount point attributes 43
    - example 43, 46
  - mount (AIX or UNIX) 16
  - mvlogin 41, 50, 56, 115, 117, 138
  - mvlogout 41, 48, 70, 115
  - mvlogout 48, 51
  - net use 21, 49, 50, 51
  - net use (DOS) 16
  - nfsstat 62
  - rm 24, 35
  - rm (UNIX) 80
  - rmdir 24, 25, 109
  - showattr 23, 33, 41, 42, 45, 51, 56, 67, 115, 117
    - displaying default and mount point attributes 44
    - displaying site and mount point attributes 22
  - showmount 65
  - tso allocate 59
  - tso help mount 53
  - tso help unmount 53
  - tso mount 58
  - umount 47, 49
  - unmount 53, 68, 113, 125
  - vi 10, 22
- command (AIX, UNIX) 39
- command syntax
  - AIX user 39
  - DFSMS/MVS user 53
- commit parameter 32
- commit procedure 11
- compatibility, POSIX 31
- control files 4
- copy command 26
- cp command 21
- creating a mount point for a PDS 25
- creating a PDS 24
- creating a PDSE 24
- creating an MVS file 23
  - direct access 23
  - PDS 23
  - PDSE 23
  - physical sequential 22
- creating conventional MVS data sets 4
- creating direct access files 23
- creating external link 34
- creating physical sequential files 22
- creating VSAM files 26
- creation (data set creation) attributes 76
- credentials allocation failure, message GFSA827I 108
- crlf end-of-line specifier 13
- ctime 147
- cyls data set attribute 76

## D

- D37 ABEND 111
- data access/creation commands 10
- data conversion 58
- data conversion (delim, filedata) 59
- data management error message 111
- data set 4
  - access 4
  - attribute defaults 4
  - attributes 4
  - cataloged 3, 15
  - creating 4
  - default attributes 4
  - definition 3, 14
  - export 5
  - location 4
  - organizations supported 3, 15
  - storage 4
  - structure 4
- data set creation attributes 76, 78
- data set serialization 17
- data\_sync parameter 32
- DataCaching 128
- DataCaching attribute 72, 73
- dataclas attribute 76
- DCB (data control block) parameters 76
- DDNAME EXPORTS 110
- DDNAME NFSATTR 97
- default attributes, displaying 45
- default attributes, overriding 43, 57
- defaults, displaying 45, 67
- delay parameter 68
- DelayWrite attribute 72
- DelayWrite keyword 128
- deleting
  - migrated files 81
  - tasks 101
- Delim attribute 72
- delim keyword 129
- delim parameter 59
- DFSMS/MVS command syntax 53
- DFSMSshm 81
- dir command 137
- dir data set attribute 76
- direct access (DA) 144, 147
- direct access (DA) files 21
- disconnecting a mount point 47, 68
- displaying client statistical information 62
- displaying default and mount point attributes 44, 45, 67
- displaying default attributes 45, 68
- displaying remote server mount information 65
- Domain Name Server 107
- DOS client, Network File System 13
- double byte character set (DBCS) 58
- dsntype attribute 100
- dsntype data set attribute 76
- dsntype(library) attribute 24
- dsntype(pds) attribute 24
- dsorg attribute 98
- dsorg(da) attribute 23

- dsorg(da) parameter 23
- dsorg data set attribute 76
- dsorg(ps) example, attribute 22
- dtpref value 18, 36
- duration parameter 68
- dynamic allocation failure 108, 109
- DynamicSizeAdj attribute 72

## E

- E37 ABEND 111
- EAGAIN, EACCESS 37
- EBCDIC to ASCII conversion 79
- end-of-line (EOL) terminator 112
- end-of-line specifiers 12, 79
- ending your MVS session 48, 70
- entry-sequenced data set (ESDS) 21, 26
- errno
  - EAGAIN, EACCESS 37
  - EWOLDBLOCK 38
- error messages 93
- ESTAE exit routine 102
- estae macro 132
- etc/passwd 117
- etc/passwd file 115
- EWOLDBLOCK 38
- executebitoff processing attribute 79
- executebiton processing attribute 79
- export spanning pathnames 7
- Export Spanning Pathnames 101
- exportfs operand 43, 114
- exports data set 5, 138
- extended binary-coded decimal interchange code (EBCDIC) 12, 30
- extended format data sets, exploiting 27
- extending PDS or PDSE member 25
- External Data Representation (XDR) 2
- external link 34
- extlink attribute 32
- extlink processing attribute 80

## F

- fastfilesize processing attribute 80, 144
- fbytes value 18
- ffiles value 18
- file creation attributes 76
- File exists message 135
- file extension mapping 15
- file naming conventions 21
- file security 117
- file size determination 10, 143
- file\_sync parameter 32
- file system 3, 14
- file system size 18, 36
- file timeout 104
- File Usage Block 111
- file usage block, message GFSA881I 111
- filedata=binary operand 59
- filedata=not specified operand 60
- filedata parameter 59

- filedata=text operand 59
- fileextmap attribute 15
- fileextmap processing attribute 80
- files, locking 37
- files (MVS), saving 12
- filesystem operand 57
- filesystype parameter 130
- fixed-blocked file format 75
- flushing partial records, message GFSA811I 105
- fn\_delimiter site attribute 87
- freeze=off operand 44, 112
- freeze=offhfs operand 44
- freeze=on operand 44, 112
- freeze=onhfs operand 44
- French EBCDIC format (CCSID 297) 58
- FSF\_CANSETTIME, properties 18
- FSF\_HOMOGENEOUS, properties 18
- FSF\_LINK, properties 18
- FSF\_SYMLINK, properties 18
- fsinfo, static file system 18, 36
- fsstat, dynamic file system 18

## G

- getattr 100
- getattr operation 78
- group number (GID) 10, 33

## H

- HFS() keyword 89
- hfs attribute 32
- HFS file system 29
- hfs(prefix) site attribute 88
- HFS site attribute 33
- hfsbtimeout site attribute 88
- hierarchical file system (HFS) 2, 29, 33, 34, 35, 84, 101, 104, 105
- high-level qualifier (HLQ) 3
- high-level qualifiers 14
- hostname
  - mvslogin command 56

## I

- IDCAMS error 109
- installation default settings, overriding 43
- installing commands 41, 55
- integrated catalog facility (ICF) 3, 15
- Integrated Catalog Facility (ICF) 3
- inter-process communication (IPC) queue 103
- invarsec value 18
- ISHELL utility 30

## J

- job control language (JCL) 76, 110

## K

- Kerberos 126
- key-sequenced data set (KSDS) 21, 26, 96
- keys data set attribute 77

## L

- leadswitch site attribute 88
- If end-of-line specifier 13
- line terminators 12, 13
- link request 17
- linkmax value 18, 36
- list=dsnames operand 113, 114
- list=mounts operand 105, 113, 114
- In command 34
- localpath 40
- lockf function 37
- locking a file 38
- locking records 37
- log data set 5
- logicalcache attribute 97
- logicalcache site attribute 45, 68, 88
- logon and file security 117
- logout, recovering from automatic 44, 62
- logout attribute 42
- logout site attribute 45, 68, 88
- lookup operation 78
- lookup request 7, 101
- Irecl data set attribute 77
- ls (UNIX) command 79, 80, 144
- ls command 25

## M

- mapleaddot processing attribute 80
- maplower attribute 9, 12
- maplower processing attribute 80
- mapped processing attribute 80
- maxfileproc parameter 133
- maxfilesize value 18
- maxrdforszleft site attribute 45, 68, 88
- maxsockets 104
- maxtimeout attribute 99, 100
- maxtimeout site attribute 45, 68, 88
- messages
  - A FILE SYSTEM WITH THE SAME NAME IS ALREADY MOUNTED. 125
  - CANNOT CREATE UDP SERVICE. 103
  - CANNOT OPEN CLIENT LOG DATA SET, text1, text2. 129
  - Cannot open input file, text1:text2 message 134
  - Cannot open output file, text1:text2 message 134
  - CANNOT OPEN THE ATTRIBUTE DATA SET. 97
  - CANNOT OPEN THE CHKLIST DATA SET. 111
  - CANNOT OPEN THE EXPORTS DATA SET. 110
  - Cannot read input file, text1:text2 message 134
  - Cannot resolve local host name message 134
  - Cannot write output file, text1: text2 message 134
  - CATALOG (text) COULD NOT BE LOCATED. 110
  - CLIENT LOG DATA SET, text, FLUSHED. 129



messages (continued)

CLIENT LOG DATA SET SWITCHED TO text. 130  
 CLIENT LOG DATA SET text RE-INITIALIZED. 129  
 CLIENT LOGGING ENDED. 129  
 CLN\_CCSID AND SRV\_CCSID WILL BE IGNORED  
 AS XLAT OPTION IS OFF. 128  
 COMMAND text NOT VALID. 115  
 CREATE FAILED FOR text. 109  
 CREATED TASK(h\_digits) - text1 - text2. 101  
 DATA SET text1 CREATION USING DATA CLASS =  
 text2. 107  
 DELETING TASK(h\_digits) - text. 101  
 Directory text not mounted. message 135  
 DYNAMIC ALLOCATION: INPUT VALIDATION  
 ROUTINE REJECTED ALLOCATION. 109  
 Enter MVS password: 117  
 Enter new MVS password: 117  
 EOL SEQUENCE MISMATCH FOR DATA SET  
 text1(text2). 112  
 Error: Can't open text for read. 117  
 Error: Directory text not mounted. 117  
 Error: Drive text not mounted. 117  
 Error: filesystem text is local. 117  
 ERROR ENCOUNTERED WHILE PARSING  
 HOSTNAME, REASON CODE reasoncd. 129  
 ERROR ENCOUNTERED WHILE PARSING MOUNT  
 PATH, REASON CODE reasoncd. 128  
 ERROR IN READING TRANSLATION TABLE,  
 text. 118  
 ERROR RETURNED TO CLIENT: RC = d\_digits  
 <text>. 97  
 ERROR WAS DETECTED IN THE EXPORTS FILE.  
 EXPORT LIST NOT REBUILT. 114  
 EXPORT LIST HAS BEEN REBUILT  
 SUCCESSFULLY. 114  
 EXPORT SPANNING PATHNAMES NOT  
 SUPPORTED 101  
 EXPORTS: DIRECTORY text WAS NOT  
 EXPORTED. 110  
 EXPORTS: NO VALID HOST NAMES IN text  
 LIST. 107  
 EXPORTS: text1 CANNOT BE EXPORTED  
 BECAUSE text2 ALREADY IS. 110  
 EXPORTS: UNEXPECTED OPTION (text)--  
 SHUTDOWN SCHEDULED. 110  
 EXPORTS: UNKNOWN HOST (text) HAS BEEN  
 IGNORED. 110  
 HFS CANNOT RESOLVE PATH NAME text. 105  
 HFS MOUNT PROCESSING ACTIVE. 113  
 HFS MOUNTS RESUMED. 105  
 HFS MOUNTS SUSPENDED. 104  
 HFS PROCESSING DISABLED. 113  
 HFS PUBLIC PATHNAME SPECIFIED BUT HFS IS  
 NOT ENABLED 101  
 Host Error: text. 117  
 Host text1 returned error d\_digits: text2 116  
 IDCAMS ERROR: text. 109  
 Input path name cannot be equal to output path  
 name. 134  
 INSTALLATION DEFAULT TRANSLATION TABLE  
 CANNOT BE INITIALIZED. 118

messages (continued)

INVALID DSNTYPE SPECIFICATION IN LINE  
 d\_digits. 100  
 INVALID DSORG SPECIFICATION IN LINE  
 d\_digits. 98  
 INVALID OPTION SPECIFICATION OF text IN LINE  
 d\_digits. 101  
 INVALID PUBLIC SPECIFICATION IN LINE  
 d\_digits 101  
 INVALID RECFM SPECIFICATION (text). 97  
 INVALID RECFM(text) - MUST SPECIFY U, F, OR  
 V. 100  
 INVALID SECURITY SPECIFICATION IN  
 d\_digits 101  
 INVALID SIDEFILE SPECIFICATION IN LINE  
 d\_digits 101  
 INVALID SMS\_keyword SPECIFICATION IN LINE  
 d\_digits. 100  
 INVALID SPECIFICATION RECFM(text). 100  
 INVALID UNIT SPECIFICATION IN LINE  
 d\_digits. 98  
 INVALID VOL SPECIFICATION IN LINE  
 d\_digits. 98  
 LOG DATA SET IS RE-INITIALIZED. 115  
 LOG DATA SET IS SWITCHED FROM text1 TO  
 text2. 115  
 LOG DATA SET text IS FLUSHED. 115  
 MISSING LEFT PARENTHESIS IN LINE  
 d\_digits. 98  
 MISSING LEFT PARENTHESIS IN text  
 KEYWORD. 127  
 MISSING RIGHT PARENTHESIS IN LINE  
 d\_digits. 98  
 MISSING RIGHT PARENTHESIS IN text  
 KEYWORD. 127  
 MOUNT FAILED: text 105  
 MOUNT HANDLE DATABASE CANNOT BE  
 READ. 95  
 MOUNT PROCESSING ACTIVE. 112  
 MOUNT PROCESSING RESUMED. 112  
 MOUNT PROCESSING SUSPENDED. 112  
 Must be a root user to issue 'character' flag  
 message 134  
 NETWORK FILE SYSTEM CLIENT DAEMON TASK  
 RESTARTED. 132  
 NETWORK FILE SYSTEM CLIENT LOGGING IS  
 TERMINATED. 131  
 NETWORK FILE SYSTEM CLIENT SHUTDOWN  
 COMPLETE. 130  
 NETWORK FILE SYSTEM CLIENT SHUTDOWN IN  
 PROGRESS. 130  
 NETWORK FILE SYSTEM IS SHORT ON  
 STORAGE. 97  
 NETWORK FILE SYSTEM SERVER ESTAE EXIT  
 UNABLE TO COMPLETE PROCESSING.  
 ABEND=abend\_code. 102  
 NETWORK FILE SYSTEM SERVER  
 INITIALIZATION FAILED: text 94  
 NETWORK FILE SYSTEM SERVER LOG text SET  
 TO FORCELOG. 115

messages (*continued*)

NETWORK FILE SYSTEM SERVER LOGGING IS  
TERMINATED. 115

NETWORK FILE SYSTEM SERVER SVC DUMP  
REQUEST FAILED. REASON=reason\_code. 102

NETWORK SERVICE IS UNAVAILABLE. 103

NEW AND OLD FILES ARE NOT MEMBERS OF  
THE SAME PDS. 109

NFSLOG1 OR NFSLOG2 DD STATEMENTS NOT  
DEFINED. 115

NO ACTIVE DATA SETS. 114

NO ACTIVE HFS DATA SETS. 105

NO ACTIVE HFS MOUNT POINTS. 105

NO ACTIVE MOUNT POINTS. 114

NO SWAP REQUEST FOR NETWORK FILE  
SYSTEM SERVER FAILED. 95

OPEN FAILED RC d\_digits FOR DATA SET  
text1(text2). 109

OPENEDITION V\_REG FAILED: RV=1, RC=h\_digit1,  
RSN=h\_digit2. 94

OS/390 NETWORK FILE SYSTEM LOCK  
MANAGER (fmid) STARTED 96

OS/390 NETWORK FILE SYSTEM LOCK  
MANAGER SHUTDOWN COMPLETE 96

OS/390 NETWORK FILE SYSTEM STATUS  
MONITOR (fmid) STARTED 96

OS/390 NETWORK FILE SYSTEM STATUS  
MONITOR SHUTDOWN COMPLETE 96

OS/390 NFS SERVER CANNOT OBTAIN NFS  
PORT 2049. 104

OS/390 UNIX REGISTRATION SUCCESSFUL. 104

OS/390 version NETWORK FILE SYSTEM  
CLIENTfmid STARTED 130

OS/390 version NETWORK FILE SYSTEM SERVER  
fmid STARTED. 97

PARSE FAILED FOR ATTRIBUTE FIELD - ILLEGAL  
KEYWORD IN LINE d\_digits. 98

PARSE FAILED IN LINE d\_digits text. 98

PARSE FAILED ON ALPHABETIC FIELD FOR text  
KEYWORD. 127

PARSE FAILED ON NUMBER FIELD IN LINE  
d\_digits. 98

PARSE FAILED ON NUMERIC FIELD FOR text  
KEYWORD. 127

Password change required by host. 116

PDS text IS NOT EMPTY. 109

PROBLEMS ENCOUNTERED PARSING THE  
CHKLIST DATA SET. 111

PUBLIC PATH CANNOT BE ESTABLISHED. 108

READ FAILED FOR NETWORK FILE SYSTEM  
CLIENT MOUNT PARAMETERS. 129

READ FAILED FOR THE ATTRIBUTES DATA  
SET. 98

READ FAILED RC d\_digits FOR DATA SET  
text1(text2). 109

REaddir ON ROOT IS NOT ALLOWED. 110

RECALL FAILED FOR MIGRATED DATA SET  
text. 95

REGISTER FOR PORT d\_digits1 PROGRAM  
d\_digits2 - VERSION d\_digits 3 - FAILED. 103

messages (*continued*)

REMOVE FAILED: RC h\_digits DSN  
text1(text2). 106

REQUEST h\_digits - FILE text NOT  
ALLOCATED. 112

REQUEST h\_digits CREDENTIALS ALLOCATION  
FAILED. 108

REQUEST h\_digits INVALID CREDENTIALS  
FLAVOR d\_digits. 108

REQUEST h\_digits INVALID MEMBERNAME FOR  
text. 108

REQUEST h\_digits PARSE FAILED FOR text. 108

REQUEST HEADER DATA BLOCK ALLOCATION  
FAILED. 103

REQUESTED MEMORY NOT AVAILABLE. 95

RETRANS OPTION WILL BE IGNORED AS HARD  
OPTION IS ON. 128

Retype new MVS password: 117

Retyped password does not match 116

SERVER SHUTDOWN COMPLETE. 95

SERVER SHUTDOWN IN PROGRESS. 95

SERVICE REQUESTER DOES NOT HAVE  
SECURITY PRIVILEGE. 105

SMF PROCESSING ACTIVE FOR FILE  
TIMEOUT. 104

SMF PROCESSING ACTIVE FOR USER  
LOGOUT. 104

SMF PROCESSING SUSPENDED FOR FILE  
TIMEOUT. 104

SMF PROCESSING SUSPENDED FOR USER  
LOGOUT. 104

SUBTASK TERMINATED: h\_digits. 102

TASK h\_digits1 TCB h\_digits2 PROGRAM = text1 =  
text2. 112

text. 97, 113

text : can't find name for uid d\_digits. 115

text: Error: cannot determine server. 116

text: Error: unknown return from usage routine. 116

text ACTIVE = d\_digits. 113

text DEALLOCATED. 113

text IS AN INCORRECT KEYWORD FOR MOUNT  
PARAMETER. 128

text IS NOT A VALID DATA SET NAME. 113

text IS NOT A VALID MEMBER NAME. 114

text logged in ok. 116

text NOT ALLOCATED. 113

text NOT MOUNTED. 113

text UNMOUNTED. 113

text1: 117

text1: Error: text2 mounted from server text3, not  
text4. 116

text1: host "text2" unknown. 116

text1: text2 116, 117

text1(d\_digits1) IS SET TO THE DEFAULT VALUE,  
text1(d\_digits2,d\_digits3,d\_digits4). 101

text1 FAILED FOR text2, text3. 130

text1(text2) 113

text1(text2) ACTIVE = d\_digits. 113

text1 UNSUPPORTED DSORG text2. 108

THE ATTRIBUTE VALUE d\_digits1 IS NOT IN THE  
RANGE OF d\_digits2 TO d\_digits3. 99

messages (continued)

THE FILE SIZE IS TOO LARGE. FN=text,  
 OFFSET=h\_digit1, LEN=h\_digit2. 96  
 THE IP ADDRESS OF REMOTE HOST NAME,  
 hostname, COULD NOT BE RESOLVED. 126  
 time\_stamp. 97  
 TRANSLATION TABLE text IS LOADED. 118  
 uid text logged out ok. 116  
 UNABLE TO BIND A RESERVED PORT TO  
 SOCKET socketnum. 132  
 UNABLE TO CREATE IPC QUEUE. 103  
 UNABLE TO SETUP ERROR RECOVERY (ESTAE),  
 RETURN CODE returncd. 132  
 UNEXPECTED END OF STRING ON END OF  
 PARSE IN LINE d\_digits. 98  
 UNEXPECTED ERROR DETECTED: d\_digits  
 text. 97  
 Unknown flag '-character' 115  
 Unknown flag '-character' message 134  
 Unknown host text message 134  
 UNKNOWN KEYWORD ENCOUNTERED AROUND  
 POSITION d\_digit. 128  
 UNMOUNT COMMAND FAILED: MOUNT POINT  
 STILL IN USE. 113  
 UNREGISTER PROGRAM d\_digits1 VERSION  
 d\_digits2 - FAILED. 103  
 usage: text [-pn] [-g group][-a account] hostname  
 [mvs\_username] 116  
 usage: text[-a] [-d] [-e] [host] 133  
 usage: text hostname 116  
 usage: text input output message 134  
 usage: text1 [-t] hostname [text2] 117  
 VERIFY: (text) IS NOT A VSAM DATA SET. 114  
 VERIFY FAILED WITH RC = d\_digits FOR  
 (text). 114  
 VERIFY SUCCESSFUL FOR (text). 114  
 WRITE FAILED RC d\_digits FOR DATA SET  
 text1(text2). 110

messages, client

GFSC100E 124  
 GFSC101E 124  
 GFSC102E 124  
 GFSC103E 124  
 GFSC105E 124  
 GFSC106E 124  
 GFSC107E 124  
 GFSC110E 125  
 GFSC200E 125  
 GFSC201E 125  
 GFSC202E 125  
 GFSC203E 125  
 GFSC204E 125  
 GFSC205E 125  
 GFSC206E 125  
 GFSC207E 126  
 GFSC208E 126  
 GFSC209E 126  
 GFSC210E 126  
 GFSC211E 126  
 GFSC212E 126  
 GFSC213E 126

messages, client (continued)

GFSC214E 127  
 GFSC300E 127  
 GFSC301E 127  
 GFSC302E 127  
 GFSC303E 127  
 GFSC304E 127  
 GFSC305E 127  
 GFSC307E 128  
 GFSC308E 128  
 GFSC309E 128  
 GFSC310I 128  
 GFSC311I 128  
 GFSC312I 128  
 GFSC313I 128  
 GFSC315E 128  
 GFSC317E 129  
 GFSC318E 129  
 GFSC319E 129  
 GFSC320E 129  
 GFSC500I 129  
 GFSC501I 129  
 GFSC502E 129  
 GFSC503E 129  
 GFSC504I 130  
 GFSC505E 130  
 GFSC506E 130  
 GFSC700I 130  
 GFSC701I 130  
 GFSC702I 130  
 GFSC703E 130  
 GFSC704E 130  
 GFSC705E 130  
 GFSC707E 131  
 GFSC708E 131  
 GFSC709E 131  
 GFSC710E 131  
 GFSC711E 131  
 GFSC712E 131  
 GFSC713E 131  
 GFSC714E 132  
 GFSC715E 132  
 GFSC716I 132  
 GFSC721E 132  
 GFSC722E 132  
 GFSC723E 132  
 GFSC724E 132  
 GFSC725E 133  
 GFSC726E 133  
 GFSC727W 133  
 GFSC728E 133  
 GFSC729W 133  
 GFSC840I 133  
 GFSC841E 134  
 GFSC843E 134  
 GFSC845I 134  
 GFSC846E 134  
 GFSC847E 134  
 GFSC848E 134  
 GFSC849E 134  
 GFSC850E 134

messages, client (*continued*)

GFSC854I 134  
GFSC855E 134  
GFSC856E 135  
GFSC858E 135

messages, client platform (AIX)

Cross device link message 135  
Directory Not Empty message 135  
File Name Too Long message 135  
I/O Error (with possible system programmer response) 135  
I/O Error (with possible user response) 136  
Is a directory 136  
No space left on device 137  
No such device 137  
No such file or directory 137  
Not a directory 137  
Not Owner 137  
Permission denied 137  
Read Only File System 138  
Stale NFS File Handle 138  
Weak Authorization 138

messages, console

CANNOT CREATE UDP SERVICE. 103  
CANNOT OPEN THE ATTRIBUTE DATA SET. 97  
CANNOT OPEN THE CHKLIST DATA SET. 111  
CANNOT OPEN THE EXPORTS DATA SET. 110  
ERROR WAS DETECTED IN THE EXPORTS FILE.  
EXPORT LIST NOT REBUILT. 114  
EXPORTS: DIRECTORY text WAS NOT  
EXPORTED. 110  
EXPORTS: NO VALID HOST NAMES IN text  
LIST. 107  
EXPORTS: text1 CANNOT BE EXPORTED  
BECAUSE text2 ALREADY IS. 110  
EXPORTS: UNEXPECTED OPTION (text)--  
SHUTDOWN SCHEDULED. 110  
EXPORTS: UNKNOWN HOST (text) HAS BEEN  
IGNORED. 110  
HFS MOUNT PROCESSING ACTIVE. 113  
HFS MOUNTS RESUMED. 105  
HFS MOUNTS SUSPENDED. 104  
HFS PROCESSING DISABLED. 113  
INSTALLATION DEFAULT TRANSLATION TABLE  
CANNOT BE INITIALIZED. 118  
LOG DATA SET IS RE-INITIALIZED. 115  
LOG DATA SET IS SWITCHED FROM text1 TO  
text2. 115  
LOG DATA SET text IS FLUSHED. 115  
MOUNT HANDLE DATABASE CANNOT BE  
READ. 95  
MOUNT PROCESSING ACTIVE. 112  
MOUNT PROCESSING RESUMED. 112  
MOUNT PROCESSING SUSPENDED. 112  
NETWORK FILE SYSTEM SERVER ESTAE EXIT  
UNABLE TO COMPLETE PROCESSING.  
ABEND=abend\_code. 102  
NETWORK FILE SYSTEM SERVER  
INITIALIZATION FAILED: text 94  
NETWORK FILE SYSTEM SERVER LOG text SET  
TO FORCELOG. 115

messages, console (*continued*)

NETWORK FILE SYSTEM SERVER LOGGING IS  
TERMINATED. 115  
NETWORK FILE SYSTEM SERVER SVC DUMP  
REQUEST FAILED. REASON=reason\_code. 102  
NETWORK SERVICE IS UNAVAILABLE. 103  
NFSLOG1 OR NFSLOG2 DD STATEMENTS NOT  
DEFINED. 115  
NO ACTIVE DATA SETS. 114  
NO ACTIVE HFS DATA SETS. 105  
NO ACTIVE HFS MOUNT POINTS. 105  
NO ACTIVE MOUNT POINTS. 114  
NO SWAP REQUEST FOR NETWORK FILE  
SYSTEM SERVER FAILED. 95  
OPENEDITION V\_REG FAILED: RV=1, RC=h\_digit1,  
RSN=h\_digit2. 94  
OS/390 NFS SERVER CANNOT OBTAIN NFS  
PORT 2049. 104  
OS/390 version NETWORK FILE SYSTEM SERVER  
fmid STARTED. 97  
PARSE FAILED IN LINE d\_digits text. 98  
PROBLEMS ENCOUNTERED PARSING THE  
CHKLIST DATA SET. 111  
PUBLIC PATH CANNOT BE ESTABLISHED. 108  
READ FAILED FOR THE ATTRIBUTES DATA  
SET. 98  
RECALL FAILED FOR MIGRATED DATA SET  
text. 95  
REGISTER FOR PORT d\_digits1 PROGRAM  
d\_digits2 - VERSION d\_digits 3 - FAILED. 103  
SERVER SHUTDOWN COMPLETE. 95  
SERVER SHUTDOWN IN PROGRESS. 95  
SMF PROCESSING ACTIVE FOR FILE  
TIMEOUT. 104  
SMF PROCESSING ACTIVE FOR USER  
LOGOUT. 104  
SMF PROCESSING SUSPENDED FOR FILE  
TIMEOUT. 104  
SMF PROCESSING SUSPENDED FOR USER  
LOGOUT. 104  
TASK h\_digits1 TCB h\_digits2 PROGRAM = text1 =  
text2. 112  
text. 113  
text ACTIVE = d\_digits. 113  
text DEALLOCATED. 113  
text IS NOT A VALID DATA SET NAME. 113  
text IS NOT A VALID MEMBER NAME. 114  
text NOT ALLOCATED. 113  
text NOT MOUNTED. 113  
text UNMOUNTED. 113  
text1(d\_digits1) IS SET TO THE DEFAULT VALUE,  
text1(d\_digits2,d\_digits3,d\_digits4). 101  
text1(text2) 113  
text1(text2) ACTIVE = d\_digits. 113  
UNABLE TO CREATE IPC QUEUE. 103  
UNMOUNT COMMAND FAILED: MOUNT POINT  
STILL IN USE. 113  
UNREGISTER PROGRAM d\_digits1 VERSION  
d\_digits2 - FAILED. 103  
VERIFY: (text) IS NOT A VSAM DATA SET. 114

messages, console *(continued)*

VERIFY FAILED WITH RC = d\_digits FOR  
(text). 114

VERIFY SUCCESSFUL FOR (text). 114

messages, server

GFSA320I(<PROCNAME>) 94

GFSA322I(<PROCNAME>) 94

GFSA323I 94

GFSA324I 94

GFSA325I 95

GFSA328I(<PROCNAME>) 95

GFSA329I(<PROCNAME>) 95

GFSA330I(<PROCNAME>) 95

GFSA331E(<PROCNAME>) 95

GFSA333I(<PROCNAME>) 95

GFSA334I(<PROCNAME>) 95

GFSA335E(<PROCNAME>) 95

GFSA335I 96

GFSA336E(<PROCNAME>) 96

GFSA336I 96

GFSA337I 96

GFSA338I 96

GFSA344I 96

GFSA345I 96

GFSA346I 97

GFSA347I 97

GFSA348I(<PROCNAME>) 97

GFSA349I 97

GFSA360I 97

GFSA361I 97

GFSA362I(<PROCNAME>) 97

GFSA400I 97

GFSA401I(<PROCNAME>) 97

GFSA402I(<PROCNAME>) 98

GFSA403I 99

GFSA403I(<PROCNAME>) 98

GFSA404I 98

GFSA405I 98

GFSA406I 98

GFSA407I 98

GFSA408I 98

GFSA409I 98

GFSA411I 98

GFSA412I 98

GFSA413I 99

GFSA414I 99

GFSA415I 99

GFSA416I 99

GFSA417I 99

GFSA418I 99

GFSA419I 99

GFSA420I 99

GFSA421I 100

GFSA422I 100

GFSA423I 100

GFSA424I 100

GFSA425I 100

GFSA426I 100

GFSA429I 100

GFSA430I 100

GFSA431I 84, 101

messages, server *(continued)*

GFSA434I(<PROCNAME>) 101

GFSA435E 101

GFSA436I 101

GFSA437I 101

GFSA438I 101

GFSA439I 101

GFSA440I 101

GFSA450I 101

GFSA451I 101

GFSA452I 102

GFSA470I(<PROCNAME>) 102

GFSA471I(<PROCNAME>) 102

GFSA480I 102

GFSA481I 102

GFSA482I 102

GFSA483I 102

GFSA501I 103

GFSA502I 103

GFSA554I(<PROCNAME>) 103

GFSA555I 103

GFSA556I(<PROCNAME>) 103

GFSA557I 103

GFSA558I(<PROCNAME>) 103

GFSA559I(<PROCNAME>) 103

GFSA564I(<PROCNAME>) 103

GFSA565I(<PROCNAME>) 104

GFSA566I(<PROCNAME>) 104

GFSA750I(<PROCNAME>) 104

GFSA751I(<PROCNAME>) 104

GFSA752I(<PROCNAME>) 104

GFSA753I(<PROCNAME>) 104

GFSA754I(<PROCNAME>) 104

GFSA770I 104

GFSA771I(<PROCNAME>) 104

GFSA772I(<PROCNAME>) 105

GFSA776I 105

GFSA777I 105

GFSA782I(<PROCNAME>) 105

GFSA783I(<PROCNAME>) 105

GFSA784I 105

GFSA786I 105

GFSA801I 105

GFSA802E 106

GFSA811I 106

GFSA812I 106

GFSA814I 106

GFSA815I 106

GFSA816I 107

GFSA817I 107

GFSA818I(<PROCNAME>) 107

GFSA819I 107

GFSA820I 107

GFSA821I 107

GFSA822I 108

GFSA823I(<PROCNAME>) 108

GFSA827I 108

GFSA829I 108

GFSA832I 108

GFSA833I 108

GFSA840I 108

messages, server *(continued)*

GFSA842I 108  
 GFSA843I 109  
 GFSA847I 109, 114  
 GFSA848I 109  
 GFSA849I 109  
 GFSA853I 108, 109  
 GFSA854I 108, 109  
 GFSA858I 109  
 GFSA859I 109  
 GFSA860I 110  
 GFSA862I 110  
 GFSA863I 110  
 GFSA864I(<PROCNAME>) 110  
 GFSA865I(<PROCNAME>) 110  
 GFSA866I(<PROCNAME>) 110  
 GFSA867I(<PROCNAME>) 110  
 GFSA868I(<PROCNAME>) 110  
 GFSA869I(<PROCNAME>) 110  
 GFSA871I 111  
 GFSA876I 111  
 GFSA877I 111  
 GFSA878I(<PROCNAME>) 111  
 GFSA879I(<PROCNAME>) 111  
 GFSA881I 111  
 GFSA886I 111  
 GFSA895I 112  
 GFSA896I 112  
 GFSA897I 112  
 GFSA898I 112  
 GFSA899I 112  
 GFSA900I(<PROCNAME>) 112  
 GFSA901I(<PROCNAME>) 112  
 GFSA902I(<PROCNAME>) 112  
 GFSA903I(<PROCNAME>) 112  
 GFSA904I(<PROCNAME>) 113  
 GFSA908I(<PROCNAME>) 113  
 GFSA909E(<PROCNAME>) 113  
 GFSA910I(<PROCNAME>) 113  
 GFSA911I(<PROCNAME>) 113  
 GFSA912I(<PROCNAME>) 113  
 GFSA913I(<PROCNAME>) 113  
 GFSA914I(<PROCNAME>) 113  
 GFSA915I(<PROCNAME>) 113  
 GFSA916I(<PROCNAME>) 113  
 GFSA917I(<PROCNAME>) 113  
 GFSA918I(<PROCNAME>) 113  
 GFSA919I(<PROCNAME>) 114  
 GFSA920I(<PROCNAME>) 114  
 GFSA921I(<PROCNAME>) 114  
 GFSA922I(<PROCNAME>) 114  
 GFSA923I(<PROCNAME>) 114  
 GFSA924I(<PROCNAME>) 114  
 GFSA925I(<PROCNAME>) 114  
 GFSA926I(<PROCNAME>) 114  
 GFSA927I(<PROCNAME>) 114  
 GFSA930I(<PROCNAME>) 115  
 GFSA931I(<PROCNAME>) 115  
 GFSA932I(<PROCNAME>) 115  
 GFSA933I(<PROCNAME>) 115  
 GFSA934I(<PROCNAME>) 115

messages, server *(continued)*

GFSA935I(<PROCNAME>) 115  
 GFSA936I(<PROCNAME>) 115  
 GFSA949I 115  
 GFSA950I 115  
 GFSA951I 115  
 GFSA952I 116  
 GFSA953I 116  
 GFSA954I 116  
 GFSA955I 116  
 GFSA956I 116  
 GFSA957I 116  
 GFSA958I 116  
 GFSA959I 116  
 GFSA960I 116  
 GFSA961I 116  
 GFSA964I 116  
 GFSA965I 116  
 GFSA966I 116  
 GFSA967I 117  
 GFSA968I 117  
 GFSA969I 117  
 GFSA970I 117  
 GFSA971I 117  
 GFSA972I 117  
 GFSA973A 117  
 GFSA974A 116, 117  
 GFSA975A 117  
 GFSA975I 116  
 GFSA976I 117  
 GFSA977I 117  
 GFSA978I 117  
 GFSA991E(<PROCNAME>) 117  
 GFSA996E(<PROCNAME>) 118  
 GFSA997I 118  
 GFSA998I 118  
 text1 REQUEST NOT VALID ON ALIAS NAME  
 text2. 107  
 mgmtclas data set attribute 77  
 migrated files 81  
 mintimeout attribute 99, 100  
 mintimeout site attribute 45, 68, 89  
 mkdir (UNIX) command 24  
 mkdir command 16, 21, 24  
 mknod request 17  
 mode operand 58  
 modify command 43, 44, 101, 102, 104, 105, 112  
 modifying file attributes 43, 57  
 monitored lock 37  
 monitored locks 37  
 more command 31  
 mount command 3, 13, 15, 21, 22, 49, 53, 57, 125  
   changing mount point attributes 43  
   changing site and mount point attributes 43  
   overriding default attributes 43  
 mount handle data set 5  
 Mount Handle Database failed 106  
 mount point  
   changing attributes 43  
   command 17  
   creating a mount point for a PDS 25

- mount point (*continued*)
  - definition 3, 15
  - disconnecting 47, 68
  - multiple 17
- mount points, saving 44, 62
- mount processing options 74
- mount TAG option 85, 86
- mounted file space 113
- mountpoint operand 58
- mtime 147, 149
- multiple data set creation attributes, specifying 75
- mv request 22
- MVS
  - accessing 40
  - password and user ID 41, 55
- MVS/ESA 31
- MVS file attributes 75
- MVS files 45
  - alias names 22
  - attributes 71
  - changing attributes 43
  - creating 21
  - line terminators 13
  - mount command 43, 57
  - mvlogin command 56
  - mvlogout command 48, 70
  - preparing to create 21
  - saving 12
  - selecting format 9
  - showattr command 44, 67
  - text, binary, record layout 21
  - umount command 47
  - unmount command 68
- MVS programs 17
- mvlogin command 41, 50, 55, 115, 117, 138
  - authentication errors 56
  - syntax for clients 56
- mvlogout command 41, 48, 55, 115
- mvlogout command 51

## N

- name\_max value 18, 36
- naming conventions for files 21
- Native ASCII environment support 84
- native ASCII support 8
- native path 7
- net use (DOS) command 43
- net use command 21, 49, 50, 51
- Network File System Client command, text, failed, return
  - value -1 return code returned reason code reasoncd
  - message 135
- Network File System Protocol Specification, RFC
  - 1094 124
- Network Lock Manager (NLM) 37
- Network Status Monitor (NSM) 38
- NFS clients, non-390 based 85
- NFS protocol 6
- NFS servers, non-390 based 85
- nfsattr 137
- NFSC\_001 69

- NFSERR\_STALE error response 105
- nfsstat -m 135
- nfsstat command 62
- nfstasks attribute 101
- nfstasks site attribute 45, 68, 89
- NFSXLAT DD statement 118
- no space condition 115
- no\_trunc value 18, 36
- noattrtimeout attribute 100
- noattrtimeout processing attribute 78
- noblankstrip processing attribute 79
- nochecklist site attribute 87
- nofastfilesize processing attribute 80, 145
- nofileextmap attribute 15
- nofileextmap processing attribute 80
- nohfs site attribute 88
- noleadswitch site attribute 88
- nomapleaddot processing attribute 80
- nomaplower attribute 9, 12
- nomaplower processing attribute 80
- nomaxtimeout attribute 100
- nomaxtimeout site attribute 88
- non-monitored lock 37
- non-monitored locks 37
- nonspanned data set attribute 78
- nopcnsd site attribute 89
- noreadtimeout attribute 100
- noreadtimeout processing attribute 81
- noretrieve attribute 13
- noretrieve processing attribute 81
- norlse data set attribute 77
- nowritetimeout attribute 100
- nowritetimeout processing attribute 82
- number of active clients 113
- number representation 13

## O

- obtaining an MVS password 41, 55
- OEDIT editor, UNIX 25
- OEMVS311 translation table 31
- overriding default attributes 43, 57
- overview of Network File System 1

## P

- parameter
  - commit 32
  - data\_sync 32
  - delay 68
  - duration 68
  - file\_sync 32
  - stable 32
  - unstable 32
  - vers(x) 6
- parameters
  - region 95
  - xlat() 102
- parm operand 58
- parsing failure in attributes data set, message
  - GFSA403I 98

- partitioned data set (PDS) 4, 15, 16, 21, 24, 25, 26, 144, 148, 149
  - extending 25
  - timing out while writing 25
  - updating 25
  - wildcard copy restriction 25
- partitioned data set extended (PDSE) 4, 15, 16, 24, 26, 76, 148, 149
  - extending 25
  - timing out while writing 25
  - updating 25
  - wildcard copy restriction 25
- password (MVS) 41, 55
- pathconf, retrieve POSIX information 18, 36
- PCNFSD authentication request 42
- pcnfsd site attribute 89
- percentsteal site attribute 45, 68, 89
- permission bits, checking UNIX 90
- permission denied message 42, 56
- physical sequential 4
- physical sequential (PS) 147
- physical sequential (PS) files 21
- port registration failure, message
  - GFSA554I(<PROCNAME>) 103
- portable operating system interface (POSIX) 1, 31
- Portmapper 7, 103
- processing attributes 78, 83
- protecting your z/OS UNIX System Services files 33
- proto, mount request 126
- proto attribute 7, 72
- public keyword 7, 101, 108
- public(legacy\_path,hfs\_path) 89

## Q

- qualifiers 14
- qualifiers, high-level 3
- queued sequential access method (QSAM) 1, 72

## R

- random access to files 10
- rdrverf processing attribute 80
- ReadAhead attribute 72
- ReadAhead keyword 128
- readaheadmax site attribute 45, 68, 90
- readdir on root 110
- readdirplus request 17, 62
- readdirtimeout processing attribute 90
- readlink request 17
- readtimeout attribute 82, 99
- readtimeout processing attribute 81
- reason code
  - rsnc 124
- reason code reasoncd 132
- reasoncd reason code 132
- recalling migrated files 81
- recfm attribute 100
- recfm data set attribute 77
- recno, write 111
- record boundaries 12

- records, locking 37
- recordsize data set attribute 77
- recs data set attribute 76
- region parameter 95
- region size 130
- relative record 26
- relative record data set (RRDS) 21, 26
- release=dsname(member) operand 113
- release operand 113
- remote mount function 43
- remote procedure call (RPC) 2, 63, 72, 73, 103, 117, 124, 133, 138
- remove request 35
- removing a PDS or PDSE 24
- rename failure, messages GFSA814I and GFSA815I 106
- resize attribute 73
- Resource Access Control Facility (RACF) 10, 33, 105, 130, 137
- restart processing 56
- restimeout attribute 44
- restimeout site attribute 5, 90
- retc return code 124
- retrans attribute 72
- retrans option 73
- retrieve attribute 13
- retrieve(nowait) processing attribute 81
- retrieve processing attribute 81
- retrieve(wait) processing attribute 81
- retry attribute 73
- return code
  - mapping 140
  - returncd 124
  - rsnc 131
- return code 132 112
- return code retc 124
- return code returncd 124, 125, 132
- return codes 139
- returncd return code 124, 125, 132
- reuse option 26
- RFC 2055 7
- rlse data set attribute 77
- rm (UNIX) command 80
- rm command 24, 35
- rm request 22
- rmdir command 24, 25, 109
- rmdir request 22
- root 23
- root directory 2, 29
- RPC 1094 124
- rsnc reason code 124
- rsnc return code 131
- rtmax value 18, 36
- rtmult value 18, 36
- rtpref value 18, 36

## S

- saf attribute 41
- SAFEXP 6
- safexp attribute 41



- SAM striped files 27
- saving MVS files 12
- saving of mount points 44, 62
- saving PDSs, timing out 25
- sdump macro 102
- security and z/OS UNIX System Services 33
- security attribute 41
- security keyword 101
- security site attribute 90
- sequential byte stream 12
- server, z/OS NFS
  - control files 4
  - creating conventional MVS data sets 4
- server, z/OS NFS introduction 1
- server messages 93
- setattr request 17
- setownernobody processing attribute 81
- setownerroot processing attribute 81
- setting time stamps 149
- setuid|nosetuid operand 58
- sfxmax site attribute 91
- shareoptions 4
- shareoptions data set attribute 77
- shell environment 53
- showattr command 22, 23, 33, 41, 42, 45, 51, 56, 67, 115, 117
- showmount command 65
- sidefile(dsname) attribute 15
- sidefile processing attribute 81
- site attributes 86, 91
- size, file size determination 143
- smf=on 104
- smf site attribute 91
- SMS\_keyword 100
- soft|hard attribute 73
- space data set attribute 78
- spanned data set attribute 78
- srv\_ccsid 84, 128
- srv\_ccsid attribute 73
- srv\_ccsid(n) attribute 84
- srv\_ccsid(n) processing attribute 82
- srv\_ccsid parameter 8, 58, 59, 73
- stable parameter 32
- starting an MVS session 56
- statelessness 10
- stats request 17
- status operand 104, 112, 113
- STEPLIB DD statement 95
- stopping subtasks 101
- storclas attribute 78
- stow macro 106
- striped files 27
- subtype 7 104
- subtype 8 104
- Sun PC-NFS
  - syntax summary 49
- Sun PC-NFS, mvsllogin command 50
- Sun PC-NFS, mvsllogout command 51
- Sun PC-NFS, net use command 50
- Sun PC-NFS, showattr command 51
- supported clients 6

- switchlog 115
- symbolic link (symlink) 17
- symbolic links 7
- SYNAD error, message GFSA877I 111
- sync attribtue 32
- sync processing attribute 82
- synchronous write to HFS file 32
- syntax diagrams, how to read xiii
- syntax summary 49
- syntax summary (AIX, UNIX) 39
- syntax summary, z/OS command 53
- System Authorization Facility (SAF) 6, 105
- system-managed facility (SMF) 104

## T

- tag operand 58
- tag option 128
- task input/output table (TIOT) 135
- tbytes value 18
- TCB address 101
- terse option, showattr command 45, 68
- text attribute 101
- text file sharing 13
- text processing, z/OS UNIX System Services 30
- text processing attribute 79
- text processing mode 12
- tfiles value 18
- time\_delta value 18
- time\_increment 148
- time stamps 147
- timeo attribute 73
- timeout 43
- timeout attributes, interaction
  - attrtimeout attribute, interaction 82
  - readtimeout attribute, interaction 82
  - writetimeout attribute, interaction 82
- timing out while writing
  - partitioned data set (PDS) 25
  - partitioned data set extended (PDSE) 25
- translation table, for z/OS UNIX System Services 30
- Transmission control Protocol/Internet Protocol (TCP/IP) 117
- Transmission Control Protocol/Internet Protocol (TCP/IP) 1, 117, 124
- trks data set attribute 76
- tso allocate command 59
- tso help mount command 53
- tso help unmount command 53
- tso mount command 58
- tso unmount 106
- type operand 57

## U

- umount command 49
- unexpected return code, SMF 104
- Unicode 74
- Unicode Conversion Services 84
- Unicode Services 58, 102, 131
- unit data set attribute 78

- UNIX client, Network File System 39
- UNIX commands
  - mkdir 24
  - vi 43
- UNIX kernel 131
- UNIX permission bits, checking 90
- UNIX System Services
  - OpenEdition MVS 105, 117, 124, 127
- unmount (immediate, force) 69
- unmount command 53, 68, 113, 125
- unmount operand 113
- unstable parameter 32
- updating PDS or PDSE member 25
- updating your MVS password 41, 55
- USA ASCII format (CCSID 437) 58
- user datagram protocol (UDP) 103, 133
- user group (UG) 42
- user number (UID) 10, 33, 35, 42

## V

- verify operand 114
- vers, mount request 126
- vers(3) parameter 126
- vers(x) parameter 6
- VFS\_MOUNT operation 126
- VFS\_PFSCTL operation 126
- VFS\_RECOVER operation 126
- VFS\_STATFS operation 125
- VFS\_SYNC operation 125
- vi command 10, 22
- vi editor 24, 26
- virtual storage access method (VSAM) 1, 4, 21, 26, 72, 114, 147, 148
- virtual storage access method (VSAM ) 96
- vol attribute 98
- vol data set attribute 78
- volume data set attribute 78
- VSAM 145
- VSAM data sets
  - dsorg attribute 76
  - recordsize data set attribute 77
  - recs data set attribute 76
  - shareoptions data set attribute 77
  - spanned data set attribute 78
- VSAM KSDS data sets, spanned data set attribute 77

## W

- wait|nowait operand 58
- WebNFS support 7
- wildcard copy restriction 25
- write operation 10
- write to PDS 26
- writetimeout 137
- writetimeout attribute 82, 99
- writetimeout processing attribute 82
- writing
  - fixed-length records 12
  - text data files 12
- writing PDSs, timing out 25

- wsiz attribute 73
- wtxmax value 18, 36
- wtxmult value 18, 36
- wtxpref value 18, 36

## X

- XDR (External Data Representation) 2
- xlat() parameter 102
- xlat attribute 73
- xlat keyword 84
- xlat=N parameter 59
- xlat parameter 58
- xlat processing attribute 82
- xlat(Y) option 128
- xlat=Y parameter 59

## Z

- z/OS client, Network File System 53
- z/OS command, syntax summary 53
- z/OS UNIX System Services 1
  - accessing 33
  - delete external link 35
  - display external link 35
  - external link to MVS 34
  - mount command 33
  - security 33

---

# Readers' Comments — We'd Like to Hear from You

**z/OS  
Network File System  
User's Guide**

**Publication No. SC26-7419-02**

**Overall, how satisfied are you with the information in this book?**

	Very Satisfied	Satisfied	Neutral	Dissatisfied	Very Dissatisfied
Overall satisfaction	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

**How satisfied are you that the information in this book is:**

	Very Satisfied	Satisfied	Neutral	Dissatisfied	Very Dissatisfied
Accurate	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Complete	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Easy to find	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Easy to understand	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Well organized	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Applicable to your tasks	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

**Please tell us how we can improve this book:**

Thank you for your responses. May we contact you? ☐ Yes ☐ No

When you send comments to IBM, you grant IBM a nonexclusive right to use or distribute your comments in any way it believes appropriate without incurring any obligation to you.

---

Name

---

Address

---

Company or Organization

---

Phone No.



Cut or Fold  
Along Line

Fold and Tape

Please do not staple

Fold and Tape



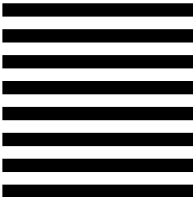
NO POSTAGE  
NECESSARY  
IF MAILED IN THE  
UNITED STATES

**BUSINESS REPLY MAIL**

FIRST-CLASS MAIL PERMIT NO. 40 ARMONK, NEW YORK

POSTAGE WILL BE PAID BY ADDRESSEE

International Business Machines Corporation  
RCF Processing Department  
M86 050  
5600 Cottle Road  
San Jose, CA U.S.A. 95193-0001



Fold and Tape

Please do not staple

Fold and Tape

Cut or Fold  
Along Line





Program Number: 5694-A01



Printed in the United States of America  
on recycled paper containing 10%  
recovered post-consumer fiber.

SC26-7419-02

